UDK 621.3:(53+54+621+66)(05)(497.1)=00 ISSN 0352-9045

INFORMACIJE MIDEM 4○2003

Strokovno društvo za mikroelektroniko
elektronske sestavne dele in materiale

Strokovna revija za mikroelektroniko, elektronske sestavne dele in materiale
Journal of Microelectronics, Electronic Components and Materials

INFORMACIJE MIDEM, LETNIK 33, ŠT. 4(108), LJUBLJANA, december 2003

MIDEM
DRUŠTVO-SOCIETY
IMAPS Slovenia Chapter

39th INTERNATIONAL CONFERENCE
ON MICROELECTRONICS, DEVICES AND MATERIALS
and the WORKSHOP on
EMBEDDED SYSTEMS

October 01.- 03.2003
Grad Ptuj, Slovenia

UDK 621.3:(53+54+621+66)(05)(497.1)=00 ISSN 0352-9045

INFORMACIJE MIDEM 4 ○ 2003

INFORMACIJE MIDEM LETNIK 33, ŠT. 4(108), LJUBLJANA, DECEMBER 2003

INFORMACIJE MIDEM VOLUME 33, NO. 4(108), LJUBLJANA, DECEMBER 2003

Revija izhaja trimesečno (marec, junij, september, december). Izdaja strokovno društvo za mikroelektroniko, elektronske sestavne dele in materiale - MIDEM.
Published quarterly (march, june, september, december) by Society for Microelectronics, Electronic Components and Materials - MIDEM.

Naslov uredništva
Headquarters

Uredništvo Informacije MIDEM
MIDEM pri MIKROIKS
Stegne 11, 1521 Ljubljana, Slovenija
tel.: + 386 (0)1 51 33 768
fax: + 386 (0)1 51 33 771
e-mail: Iztok.Sorli@guest.arnes.si
http://paris.fe.uni-lj.si/midem/

Letna naročnina znaša 12.000,00 SIT, cena posamezne številke je 3000,00 SIT. Člani in sponzorji MIDEM prejemajo Informacije MIDEM brezplačno.
Annual subscription rate is EUR 100, separate issue is EUR 25. MIDEM members and Society sponsors receive Informacije MIDEM for free.

Znanstveni svet za tehnične vede I je podal pozitivno mnenje o reviji kot znanstveno strokovni reviji za mikroelektroniko, elektronske sestavne dele in materiale. Izdajo revije sofinanci rajo Ministrstvo za znanost in tehnologijo in sponzorji društva.
Scientific Council for Technical Sciences of Slovene Ministry of Science and Technology has recognized Informacije MIDEM as scientific Journal for microelectronics, electronic components and materials.
Publishing of the Journal is financed by Slovene Ministry of Science and Technology and by Society sponsors.

Znanstveno strokovne prispevke objavljene v Informacijah MIDEM zajemamo v podatkovne baze COBISS in INSPEC.

Prispevke iz revije zajema ISI® v naslednje svoje produkte: Sci Search®, Research Alert® in Materials Science Citation Index™

Scientific and professional papers published in Informacije MIDEM are assessed into COBISS and INSPEC databases.

The Journal is indexed by ISI® for Sci Search®, Research Alert® and Material Science Citation Index™

Po mnenju Ministrstva za informiranje št.23/300-92 šteje glasilo Informacije MIDEM med proizvode informativnega značaja.

Grafična priprava in tisk Printed by	BIRO M, Ljubljana
Naklada Circulation	1000 izvodov 1000 issues

Poštnina plačana pri pošti 1102 Ljubljana
Slovenia Taxe Percue

ZNANSTVENO STROKOVNI PRISPEVKI — PROFESSIONAL SCIENTIFIC PAPERS

VSEBINA CONTENT

Obnovitev članstva v strokovnem društvu MIDEM in iz tega izhajajoče ugodnosti in obveznosti

Spoštovani,

V svojem več desetletij dolgem obstoju in delovanju smo si prizadevali narediti društvo privlačno in koristno vsem članom.Z delovanjem društva ste se srečali tudi vi in se odločili, da se v društvo včlanite. Življenske poti, zaposlitev in strokovno zanimanje pa se z leti spreminjajo, najrazličnejši dogodki, izzivi in odločitve so vas morda usmerili v povsem druga področja in vaš interes za delovanje ali članstvo v društvu se je z leti močno spremenil, morda izginil. Morda pa vas aktivnosti društva kljub temu še vedno zanimajo, če ne drugače, kot spomin na prijetne čase, ki smo jih skupaj preživeli. Spremenili so se tudi naslovi in način komuniciranja.

Ker je seznam članstva postal dolg, očitno pa je, da mnogi nekdanji člani nimajo več interesa za sodelovanje v društvu, se je Izvršilni odbor društva odločil, da stanje članstva uredi in **vas zato prosi, da izpolnite in nam pošljete obrazec priložen na koncu revije.**

Naj vas ponovno spomnimo na ugodnosti, ki izhajajo iz vašega članstva. Kot član strokovnega društva prejemate revijo »Informacije MIDEM«, povabljeni ste na strokovne konference, kjer lahko predstavite svoje raziskovalne in razvojne dosežke ali srečate stare znance in nove, povabljene predavatelje s področja, ki vas zanima. O svojih dosežkih in problemih lahko poročate v strokovni reviji, ki ima ugleden IMPACT faktor.S svojimi predlogi lahko usmerjate delovanje društva.

Vaša obveza je plačilo članarine 25 EUR na leto. Članarino lahko plačate na transakcijski račun društva pri A-banki : 051008010631192. Pri nakazilu ne pozabite navesti svojega imena!

Upamo, da vas delovanje društva še vedno zanima in da boste članstvo obnovili. Žal pa bomo morali dosedanje člane, ki članstva ne boste obnovili do konca leta 2003, brisati iz seznama članstva.

Prijavnice pošljite na naslov:

MIDEM pri MIKROIKS

Stegne 11

1521 Ljubljana

Ljubljana, april 2003

Izvršilni odbor društva

NOISE AND NON-LINEARITY AS RELIABILITY INDICATORS OF ELECTRONIC DEVICES

J. Sikula, V. Sedlakova, P. Dobis

Brno University of Technology, Brno, Czech Republic

INVITED PAPER
MIDEM 2003 CONFERENCE
01. 10. 03 - 03. 10. 03, Grad Ptuj

Abstract: An application of noise and non-linearity measurements in analysis, diagnostics and prediction of reliability of electronic devices is discussed. The sensitivity of noise and non-linearity to the device defects and other irregularities is typical feature of these methods. Conceptions of 1/f noise, burst or RTS noise, thermal noise and third harmonic voltage are described and theirs explanation is done. The results of noise and non-linearity measuring are shown. Possible reliability indicators for conducting film resistors, MOSFETs and quantum dots are presented.

Šum in nelinearni efekti kot pokazatelji zanesljivosti elektronskih komponent

Izvleček: V prispevku pokažemo uporabo meritev šuma in nelinearnosti pri analizi, diagnozi in napovedi zanesljivosti elektronskih komponent. Glavna odlika tega pristopa je ravno občutljivost šuma in nelinearnosti na napake in druge nepravilnosti znotraj komponente. Opišemo in razložimo koncept 1/f šuma, RTS šuma, termičnega šuma in tretje harmonske napetosti, kakor tudi obrazložimo rezultate meritev. Predstavimo možne pokazatelje zanesljivosti prevodnih plasti, uporov, MOSFET tranzistorjev in kvantnih lukenj.

1. Introduction

The noise spectroscopy in time and frequency domain is one of the promising methods to provide a non-destructive characterisation of semiconductor materials and devices. This applies to both active and passive components, i.e., bipolar, quantum dots and MOS structures, on one hand, and resistors and capacitors on the other. As a main diagnostic tool it is proposed to use low frequency current or voltage noise spectral density and theirs statistical distributions.

The noise in a device, correlation with reliability and why conduction noise, especially 1/f noise, is a quality indicator for devices is indicated in /1/. Here we restrict ourselves to 1/f noise as a diagnostic tool in resistance type devices. The knowledge about 1/f noise is based on experimental facts that fit into the empirical relation for the 1/f noise with the Hooge-parameter α.

There are two kinds of 1/*f* noise in electronic devices. The first type was called the fundamental 1/*f* noise. It may be associated with the 1/*f* noise that is caused by the carrier mobility fluctuations at the charge-carrier scattering by phonons /1/. The other kind of 1/*f* noise is generated by defects in the device. It was called as excess $1/f^a$ noise and it is a characteristic for detecting of imperfections and latent defect. This kind of excess $1/ f^a$ noise may be the non-equilibrium 1/*f* noise /2, 3/.

It is known that most of failures result from the latent defects created during the manufacture processes or during the operating life of the devices. The sensitivity of excess electrical noise to this kind of defects is the main reason of investigation and use of noise as a diagnostic and prediction tool in reliability physics for the semiconductor devices lifetime assessment. The noise spectral density depends on stress and damage and varies among nominally identical devices.

The sensitivity of the noise characteristics to the structure defects and other irregularities is typical feature of these methods. It is due to microphysical origin of fluctuation caused by quantum transitions of charge carriers. Noise depends on: i) perfection of the crystal structure, number of grain boundaries, point defects, linear defects, ii) surface parameters and iii) homogeneity and manufacturing quality of the device active region.

The actual reliability of electronic devices is usually less then the maximum theoretical value of reliability depending on the attained manufacture level. It may be due to irregularities in the manufacturing processes. It was observed that chemical condition of the surface could affect the magnitude of the noise spectral density.

In the present paper the application of noise spectroscopy on thin and thick conducting film resistors, MOSFETs and quantum dots and is given. The noise characteristics variation during stress and ageing are much larger than those of the DC characteristics and then our experimental studies are used for a quality and reliability assessment.

2. Characterisation of the noise

Noise generated in semiconductor sample consists from thermal noise, burst or RTS noise, generation-recombination (GR) noise, 1/*f* noise and *1/f^a* noise. There are two experimental ways to characterize fluctuations in semiconductor sample. The first is based on the analyses of stochastic process realization in time domain. In this case average value, probability density and distribution function can be obtained. The second one uses Fourier transformation of time domain realization and gives information about second order statistical moments as is noise spectral density. The stochastic process realization is then analysed in frequency domain. This method is frequently used and noise sources are characterized by frequency dependence of noise spectral density. Noise voltage, current, and power spectral densities are used. We give short description of them.

2.1 Thermal noise

All semiconductor samples always display thermal noise caused by random motion of the charge carriers and their interaction with phonons. The spontaneous fluctuations across the resistor have a white noise spectrum given by:

the voltage noise spectral density

$$S_u = 4kTR \qquad (1)$$

the current noise spectral density

$$S_I = 4kTG \qquad (2)$$

or the power noise spectral density

$$S_p = S_u / R = S_I R = 4kT \qquad (3)$$

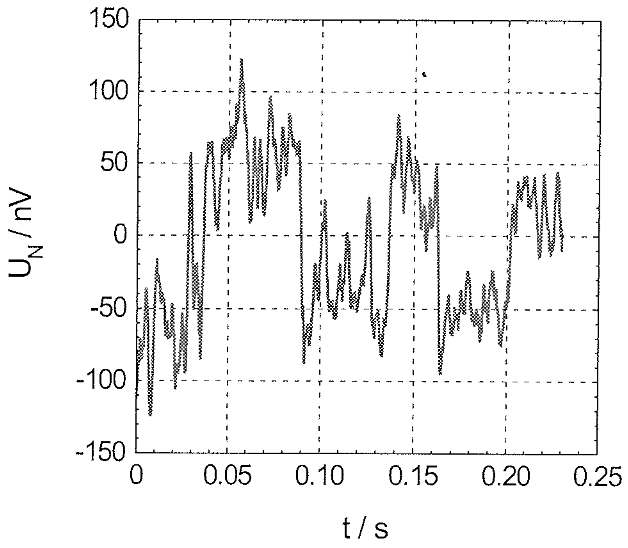


Fig.1. RTS noise voltage time dependence

where k, *T* and *R* are Boltzmann constant, the absolute temperature, and the sample resistance respectively. The thermal noise of resistors is often used to calibrate the noise- measuring set-up. The thermal noise from a biased sample is proportional to a temperature, which is larger than a substrate temperature due to Joule heating. The thermal resistance is an important indicator of delamination of the thick film layer and hence an indicator of early failures. Thus a strong increase in thermal noise under bias goes hand in hand with an increased risk of failures.

2.2 Burst or RTS noise

This type of noise is called burst or random telegraph signal (RTS) noise. It is an important indicator of a single trap activity in a small subsystem with a small number of free carriers. Such defect often appears due to laser trimming. Defect region has small dimensions and is submitted to high electric field current density. Therefore it often degrades faster and at least shows a poor noise behaviour. Time realisation of voltage fluctuation on resistor is given in Fig. 1. and noise spectral density of this type of noise for the resistor *R* = 8 kΩ is shown in Fig.2 for applied voltage 0.7 and 1.4 V.

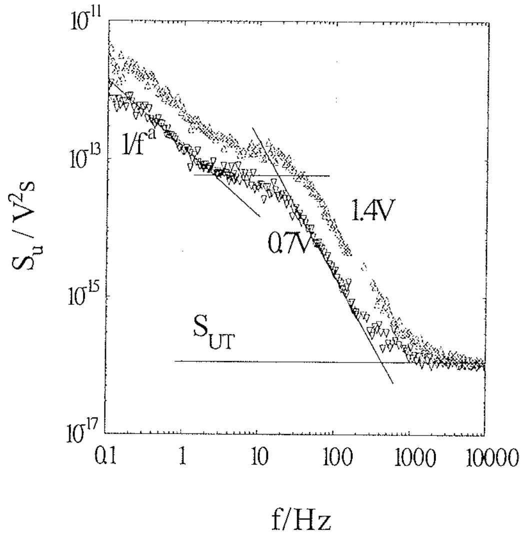


Fig.2. Noise spectral density vs. frequency

2.3 Generation-recombination noise

Generation-recombination (G-R) noise is a fluctuation in the conduction, and like burst noise is caused by the carriers number fluctuations. The amplitude density distribution is Gaussian with the Lorentzian spectrum for simple transitions between a band and traps at one energy level. Traps energy level must be near to Fermi level in order to create G-R noise. That is the reason why some devices can show G-R noise just in a certain temperature range.

2.4 Noise of 1/f type

Present theories of 1/f noise assume that there are two sources: fundamental *1/f* noise and excess *1/f^a* noise. According to Hooge /1/, *1/f* noise is due to carrier mobility fluctuations and current noise spectral density is given by a generalised Hooge's formula

$$S_I = \alpha . I^2 / f . N \qquad (4)$$

where S_I is the current noise spectral density, f is frequency, N is the total number of carriers in the sample active region, I is the device current and α is an empirical constant. Hooge constant α is now extensively used to characterise the device structure perfectness. The values of α measured on the devices ranges from 10^{-3} (poorest samples) to 10^{-7} (currently the best samples).

Normalised noise spectral density with respect to applied voltage S_u/U^2, electric current S_I/I^2 or resistance S_R/R^2 can be expressed by

$$S_u/U^2 = S_I/I^2 = S_R/R^2 = \alpha./f.N \qquad (5)$$

Normalised noise spectral density for samples with different widths and lengths 0.3mm, 0.5mm and 1mm is shown in Fig.3.

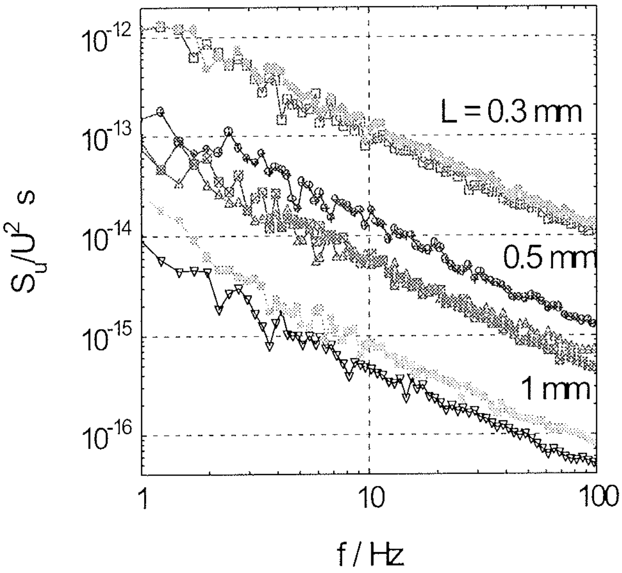


Fig. 3. Normalised voltage noise spectral density for thick film resistors

It was proposed to defined the quality and reliability indicator C_Q as

$$C_Q = S_I f / I^2 \qquad (6)$$

Normalised frequency curve of C_Q indicator is shown in Fig.4.' The better technology, the lower C_Q value and its dispersion is measured.

At present the concept of the excess noise as a quality and reliability indicator is generally used. There is also a variety of measuring set-ups and measuring conditions. Not all of them provide the best attainable resolution. Generally speaking, one has to be able to distinguish the excess noise of the device, which carries information on the device condition, from the background noise.

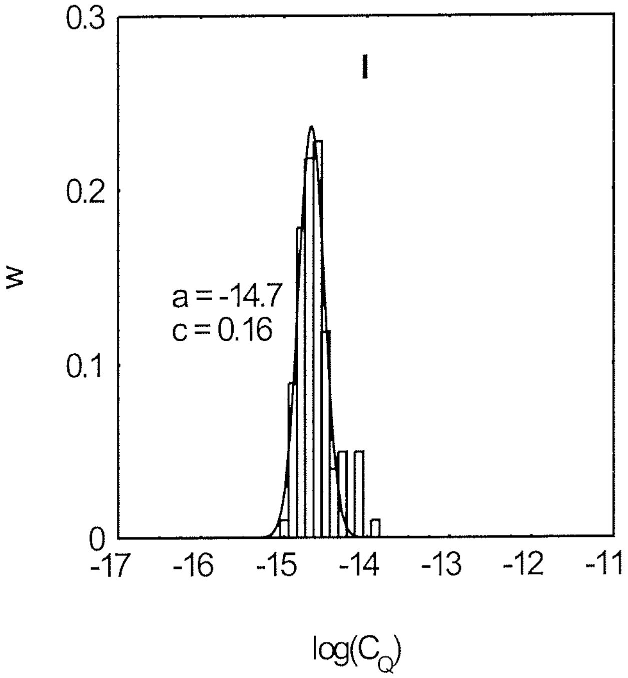


Fig.4. Normalised frequency curve of C_Q indicator

2.5 Power noise spectral density

This quantity corresponds to voltage or current fluctuation on sample resistance R. From this point of view behaviour of resistor type devices can be described by similar model as was given by Hooge for monocrystals /1/.

We propose, that power noise spectral density S_P is proportional to power dissipated by one charge carrier P_0 and inversely proportional to frequency

$$S_P = \alpha P_0 / f \qquad (7)$$

where

$$P_0 = e \mu E^2 \qquad (8)$$

α is proportionality constant, μ is charge carrier mobility, E is electric field intensity.

Total power noise spectral density including thermal noise will be given by

$$S_P = 4kT + a_\mu e E^2 / f \qquad (9)$$

where a_μ has a unit of mobility and is given by

$$a_\mu = \alpha . \mu \qquad (10)$$

Quantity a_μ does not depend on sample length and width as is shown in Fig. 6.

Power noise spectral density has lowest value $S_P = 4kT$ as is shown in Fig. 5.

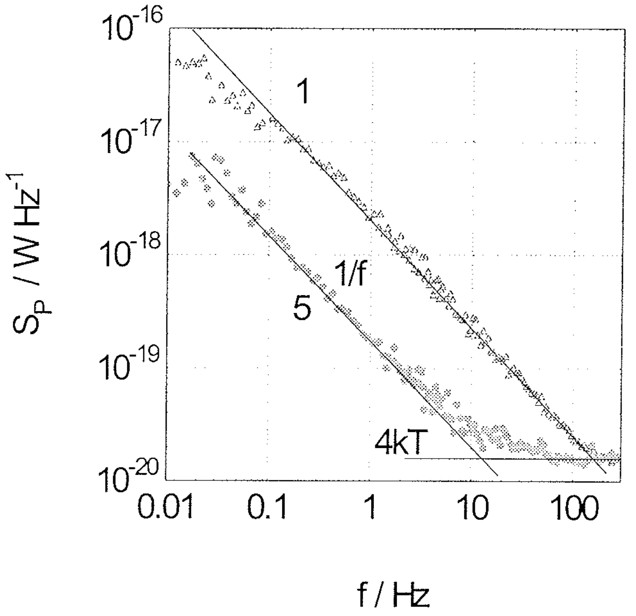


Fig. 5. Power noise spectral density for two thick film resistors

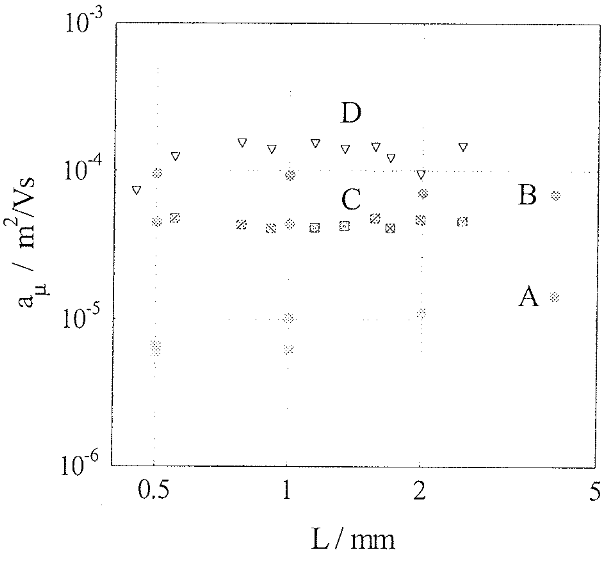


Fig.6. Quantity am for pastas 10 kΩ/ and different technologies

2.6 The non-equilibrium resistance fluctuations

The resistance of any conducting two-terminal devices *R*(*I*, *t*) is generally a non-linear function of a current *I* and can be approximated by:

$$R(I,t) = R_0(t) + \sum_{n=1}^{\infty} R_n(t)\, I^n(t) \qquad (11)$$

Here, the time dependence of coefficients $R_n(t)$ (n = 0, 1,2, ...) represents the fluctuations in the resistance. For a weakly non-linear sample the terms in (11) decrease rapidly with *n* so that for ordinary working currents the voltage fluctuations Δ*U*(*t*) across a sample under a DC current I_0, with accounting of the first three terms in (11), given by

$$\Delta U(t) = \Delta R(I,t) I_0 = \Delta R_0(t) I_0 + \Delta R_1(t) I_0^2 + \Delta R_2(t) I_0^3 \qquad (12)$$

where $\Delta U(t) = U(t) - \overline{U}$ and $\overline{U}$ is the time-averaged voltage drop across the sample. Here fluctuations of the terms in (11) are

$$\Delta R_n(t) = R_n(t) - \overline{R_n} \text{ (n = 0, 1, 2)}, \qquad (13)$$

where $\overline{R_n}$ are the time-averaged coefficients. Fluctuations $\Delta R_0(t)$ give the equilibrium $1/f^a$ noise. Fluctuations $\Delta R_1(t)$ and $\Delta R_2(t)$ are sources of the non-equilibrium resistance fluctuations. When the noise spectral density $S_U(f)$ is measured under DC current, the equilibrium and non-equilibrium components are superimposed. If the DC current is small the non-equilibrium component is hidden by the equilibrium one.

2.7. Measuring set-up

There is a variety of measuring set-ups and measuring conditions. Not all of them provide the best attainable resolution. Generally speaking, one has to be able to distinguish the excess noise of the device, which carries information on the device condition, from the background noise. To get a good measurement resolution, it is necessary to carry out the measurements in the region, where the expected noise component magnitude is distinctly higher than that of the background.

3. Non-linearity

The resistor structure consists of metallic grains and intergrain layers with semiconductor conductivity. Junctions between metal grain and semiconductor layer are non-ohmic. Important non-ohmic regions are at contacts and in vicinity of defects of resistor structure.

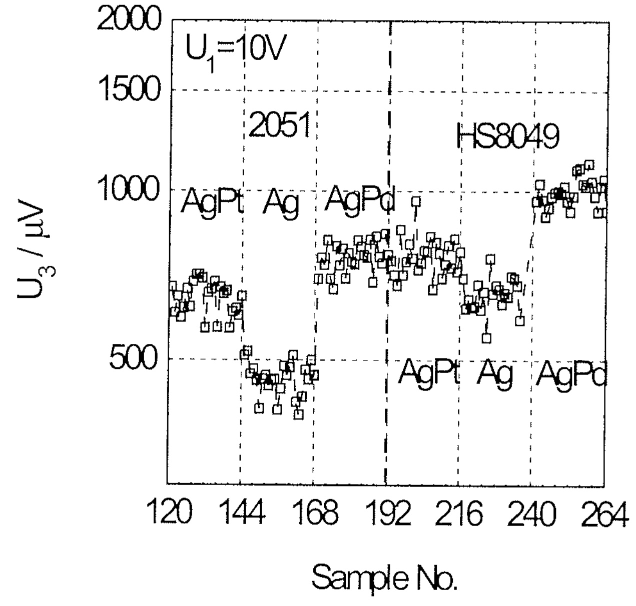


Fig. 7. Third harmonic voltage for samples 0.3x0.3mm, paste 100kΩ/□ at U_1=10V

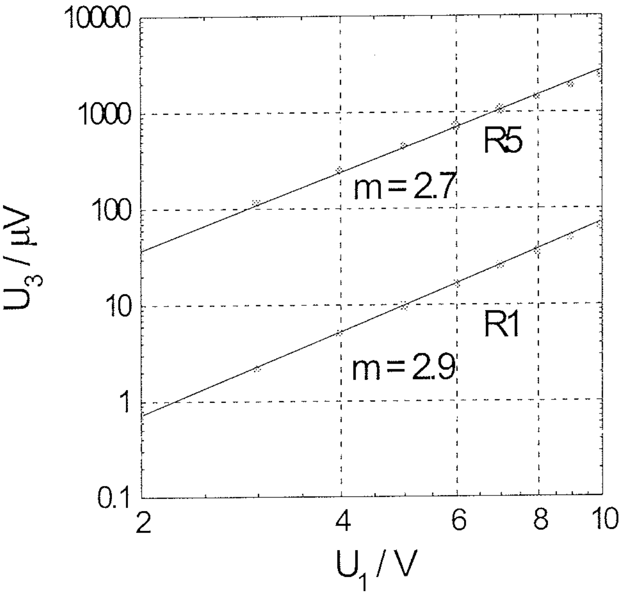


Fig.8. Third harmonic voltage as a function of first harmonic voltage

These non-ohmic regions are screened by the third harmonic voltage measurement. This non-destructive method gives additional information on the reliability of passive devices. The basic measuring frequency was 10 and 100 kHz and nonlinearity index was defined as ratio of the third harmonic and the first harmonic applied voltage, expressed in dB:

$$NLI = 20log(U_3/U_1) \qquad (14)$$

Third harmonic voltage (THV) as a function of first one U_1 was experimentally studied for different resistors and all cases the U_3 is proportional to the $U_1{}^3$ as is shown in Fig.8.

Non-linearity measurements for different resistor technologies show, that the third harmonic voltage U_3 is dependent on the type of resistor and the contact electrode material (Figs. 7.). The third harmonic voltage U_3 dependence electric field intensity E_1 or current density is shown in Figs. 9. and 10. U_3 is proportional to the third power of the electric field intensity or current density. Proportionality constant depends on the given thick film resistor technology (see curve A for AgPd and B for Ag conductors).

Absolute value of THV can be taken as reliability indicator only for devices produced by the same technology. If the resistor exhibits either a contact non-linearity or a current crowding problem then the THV will depend on current density peak. It was observed, that thick film resistors with AgPd contact electrode have higher value of third harmonic voltage, but show better long term stability and reliability comparing with Ag contact electrode resistors. In this case non-homogeneities in current density distribution are responsible for higher values of non-linearity and noise spectral density. Silver diffusion from contact electrode into the resistive layer results to the increase of conductivity of resistive layer, and then the peak of current density is shifted from contact - resistor interface into resistor volume. In this case the noise and non-linearity dominant sources are not on the contact - resistor interface, but are shifted to the thick film resistor volume. The contact - thick film interface is not so much affected by Joule heating of this spot and we suppose that irreversible processes on contact interface are weaker.

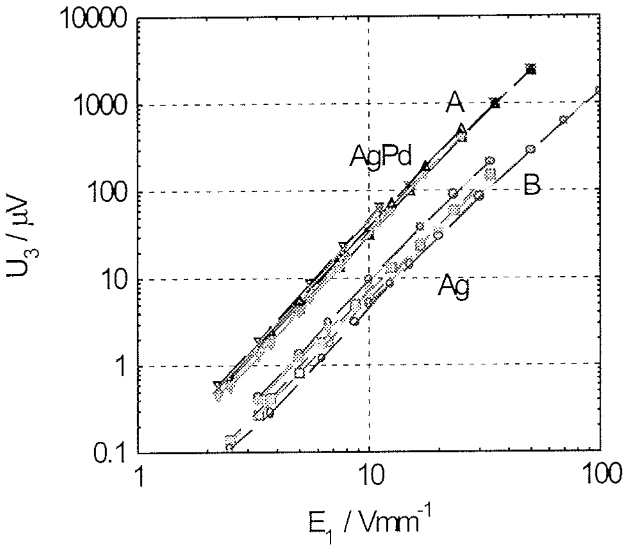


Fig. 9. THV as a function of electric field intensity for Ag and AgPd contacts

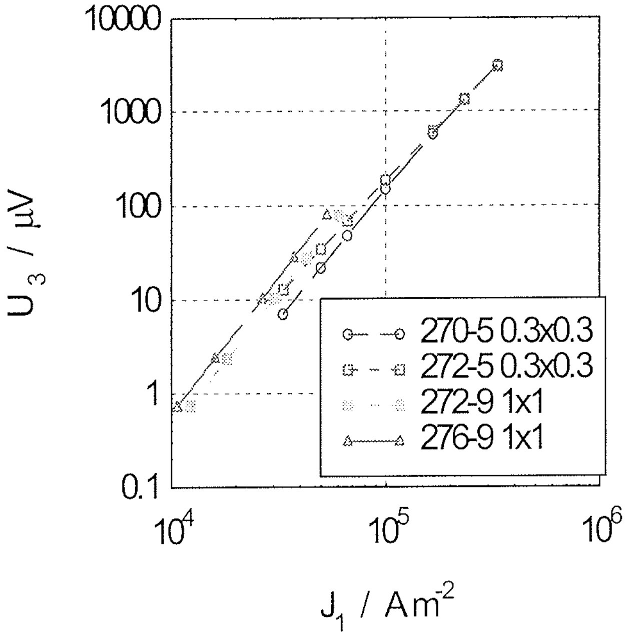


Fig.10. THV as a function of current density for Ag/Pd contacts

4. Thin and thick film resistors

Low frequency noise and non-linearity measurements are used as reliability indicators to describe ageing stability of film resistors. In order to detect technology step responsi-

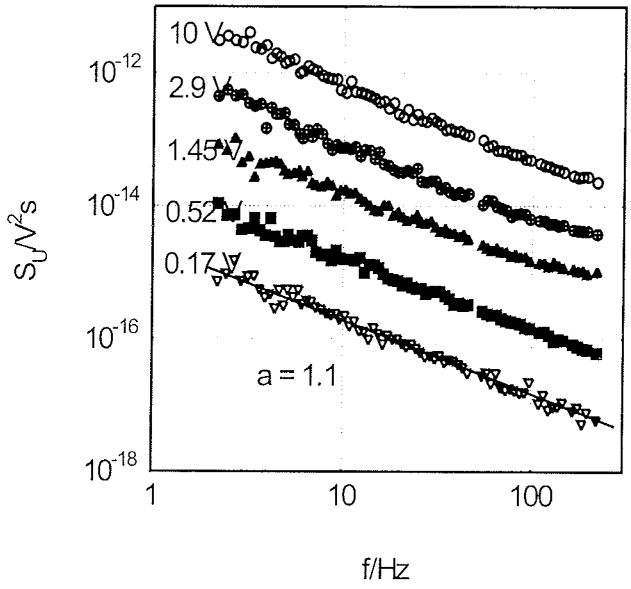


Fig. 11. Noise voltage spectral density vs. DC applied voltage for metallic resistor

ble for noise sources creation, in co-operation with some manufacturers, studies were performed at three different stages of the technology process: resistance layer deposition, laser adjustment and protective layer coating. Critical is laser adjusting operation, when cracks and defects can be generated, which are sources of noise.

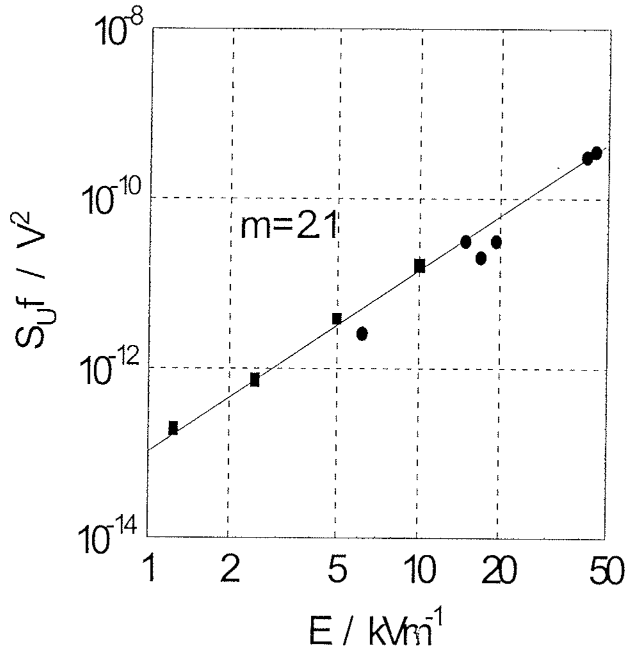


Fig. 12. 1/f noise spectral density versus electric field intensity

Noise spectral density as a function of frequency for different values of DC applied voltage of metallic resistor is shown in Fig. 11. All curves have the slopes varying from a=1.0 to 1.1. Normalized 1/f noise spectral density versus electric field intensity for thick film resistor is shown in Fig. 12. Quadratic dependence of noise spectral density on electric field intensity or applied voltage is characteristic for stationary ergodic stochastic processes.

4.1 Non-homogeneous current density distribution

The contact electrode material of the thick film resistors has important influence on their quality and reliability. Strong dependence of non-linearity and noise on the contact electrode material was observed /5/. Thick film resistors with AgPd contact electrode have higher value of third harmonic voltage, but show better long term stability and reliability comparing with Ag contact electrode resistors. We determined from the SEM analyses, that the sharpness of AgPd contact electrode is higher than Ag contact electrode one. Modelling of the current distribution for different sharpness of metallic contact cross sections was performed and it shows that the electrode geometry plays dominant role for current distribution in thick film resistor layer.

If the thick film resistor exhibits either a contact noise or a current crowding problem on a microscopic scale or both, then the calculated α from (4) will not be the α-value representing the 1/f noise properties of the thick film material. These will lead to higher α values. Sources of this excess noise are current crowding near the contacts. The *1/f* noise parameter α_a value is higher than $3\text{x}10^{-3}$.

Due to the silver diffusion conductivity between Ag contact and resistive layer varies continuously. The conductivity of resistor layer in the vicinity of contact decreases from the silver conductivity to reach the conductivity of resistive paste. For the comparison also the model without the transition region was analysed. The peak of the electric field intensity is shifted into resistor volume with respect to model without Ag diffusion (see Fig. 13.)

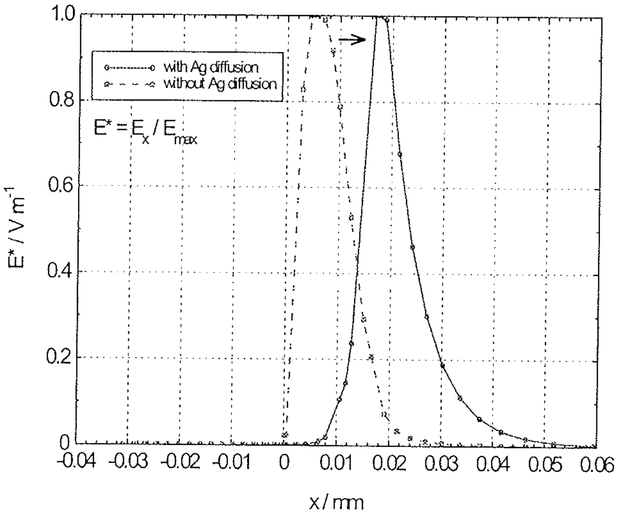


Fig. 13. Shift of electric field intensity near contact due to silver diffusion

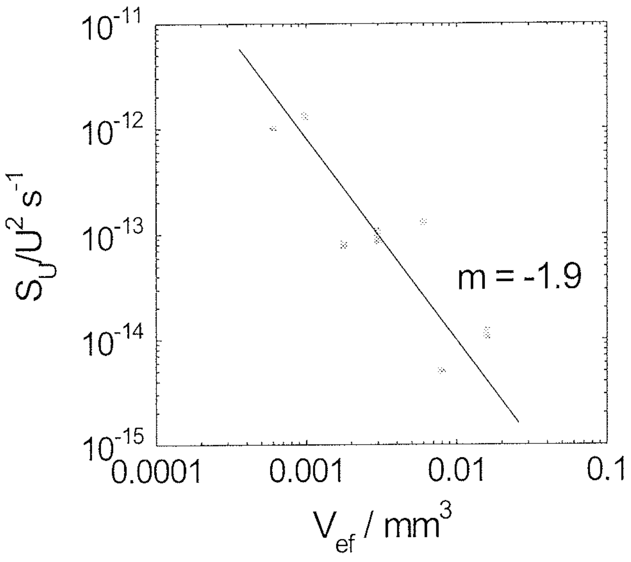


Fig.14. Normalised noise spectral density versus resistor effective volume

The silver diffusion in resistor volume causes shift of current density peak from the contact - resistor interface. This effect leads to lower stability of resistor characteristics due to ageing, but noise and non-linearity characteristics are reduced. The current density peak in the vicinity of contact affects the values of noise and non-linearity characteristics, but they are not connected with irreversible processes at the contact interface. In this case the higher value of noise or non-linearity does not indicate lower reliability.

4.2 Noise and sample volume

Normalised noise spectral density increases with decreasing sample volume. Small sample has higher value of noise spectral density. We found that noise spectral density is inversely proportional to the square of thick film sample volume (see Fig.14.). This result is not in coincidence with linear dependence predicted by Hooge empirical model /4/. We suppose that this effect is a result of non-homogeneous electric field and current crowding enhances. Decreasing the sample length the current density peak near contacts is more pronounced.

5. RTS in mosfets and quantum dots

It is supposed to be caused by individual traps, which can be either in silicon, or the oxide, or in the interface between the silicon and the oxide. Its position results on the time of the RTS pulses. For the trap located in the channel or in the boundary between the oxide and silicon at the gate side, the mean pulse time for an n-type semiconductor with a carrier concentration of 10^{16} cm^{-3}, the capture cross section 10^{16}cm^2 and a thermal velocity of 10^7 cm/s is 0.1 microseconds.

If the trap is located in the oxide then the capture τ_c and emission τ_e time constants will be longer because of tunnelling. The tunnelling time increases with the trap depth exponentially and for a depth of 1 nm the characteristic time is of the order of 1 second. The capture time τ_c is inversely proportional to the square of the drain current I_D as is shown in Fig. 15. The emphasis is on those signals showing a capture process, which deviates from the standard Shockley-Read-Hall kinetics corresponding to a quadratic dependence on the number of carriers or on the current.

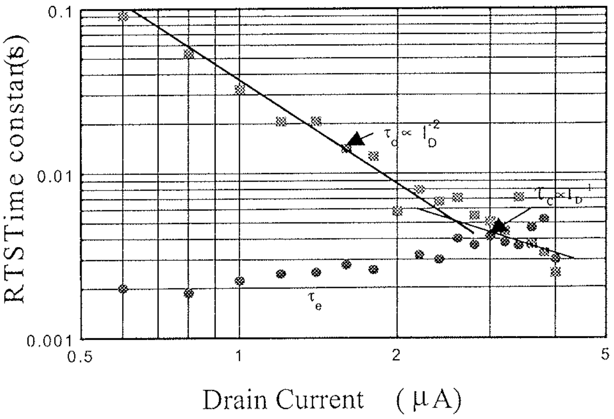


Fig. 15. Capture τ_c and emission τ_e time constants

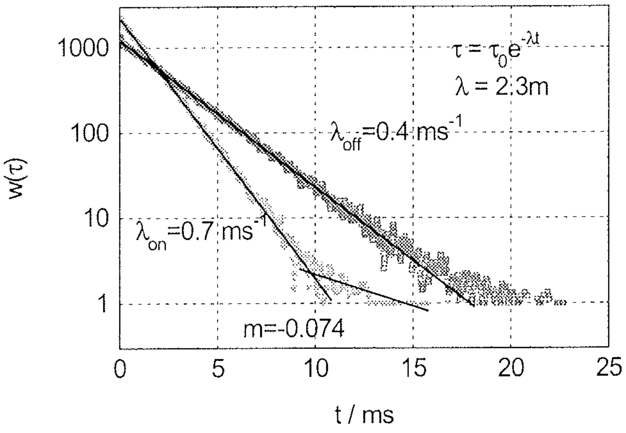


Fig. 16. Capture τ_c and emission τ_e time probability density for sample N31

Time constant distribution is exponential in first approximation only and result for a small-area Si n-MOSFET is in Fig. 16. This is second experiment on which a modified two-step approach is proposed. It includes the capture of a carrier by a trap located at the Si-SiO$_2$ interface, followed by a tunnelling process of the trapped carrier between the interface trap and a trap located in the SiO$_2$ layer. Generation-recombination process has noise spectral density Lorentzian type as is shown in Fig. 17. By this model a quadratic dependence of the capture rate on the drain current can be explained, provided that the quasi-Fermi level at the surface is below the interface trap level. Noise spectral density reach maximum value (see Fig. 18) for current at which Fermi quasi level coincide with trap energy level. From these experiments the interface trap cross-section and oxide trap cross-section can be determined. This result is based on assumption that noise is caused by charge carrier quantum transitions.

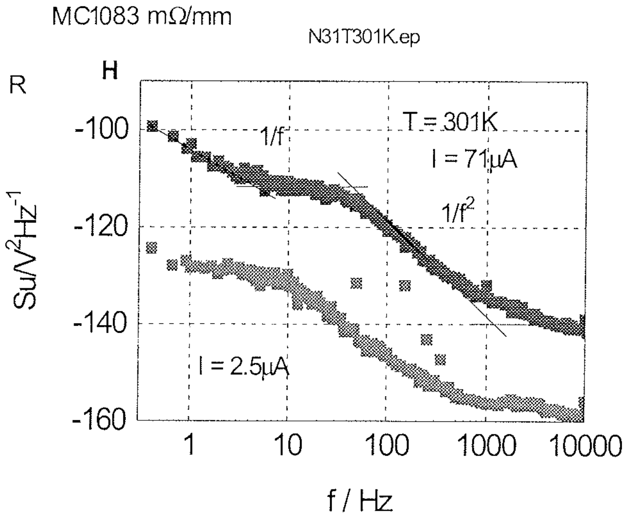


Fig. 17. Noise spectral density of n-MOSFET

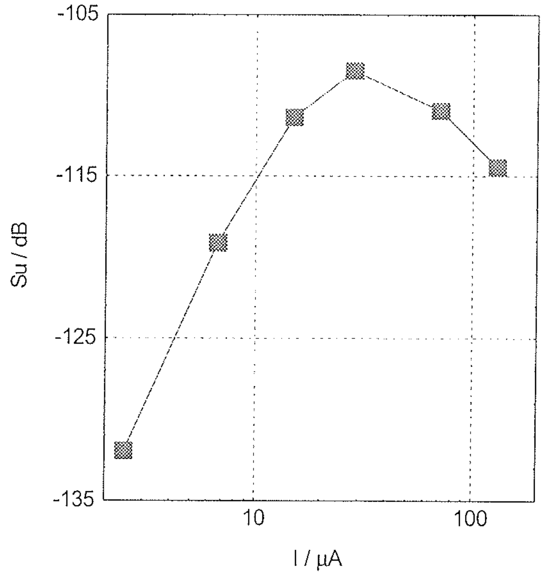


Fig. 18. *Su vs. drain current*

5.1 Quantum Dots

Quantum dot (QD) structures have attracted much attention because of interest in not only the relevance of low-dimensional electron gas physics but also future applications of high-density, low power, and highly functional integrated circuits. Tacano et al. /6/ fabricated an AlGaAs/InGaAs heterojunction field-effect transistor (FET) memory cell in a tetrahedral-shaped recess (TSR) structure, which has a hole-trapping QD as a floating gate, and succeeded in observing RTS noise in the retention characteristics of the memory cell. RTS observed in short- and narrow-channel FET are direct evidence of a single charge capture and emission. An analysis of RTS in this work quantitatively explains details of hole trapping processes in the TSR QD.

Temperature dependent RTS pulses (Fig.19.) are excited up to 130 K, and the activation energy of a hole capture and emission processes were estimated as:

Et_1 = 190 meV and Et_2 = 260 meV.

Similar value of the activation energy was found in MOS structures (see Schulz /7/)

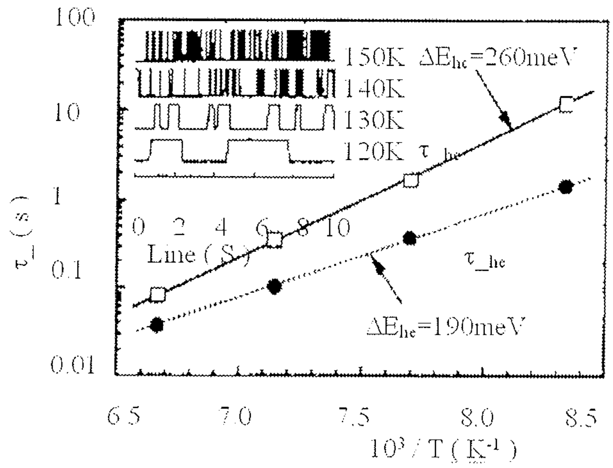


Fig. 19. RTS from QDs at various temperatures

6. Conclusion

It was proposed as a quality and reliability indicator a normalised 1/f noise spectral density and non-linearity index. Burst noise component shows that the thick and thin conducting layers are composed of chainlike structure of metal grains separated by semi-conducting or insulating layers.

Power noise spectral density of *1/f* noise is proportional to power dissipated by one charge carrier and inversely proportional to frequency. We may conclude that there are two effective carrier mobilities: one for the transport and the other for the noise characterization. RTS noise amplitude has its maximum value when the electron Imref coincides with the trap energy level. Then in quantum dots and MOS structures the RTS noise appear in a short temperature range. All models presented up to now can explain the bistability of the system, which is caused by defects in the device structure.

Acknowledgement

This research was supported by the KONTAKT Czech - Japan Government Cooperation Grant Me-605 and by the Grant MSMT: Microsyt No 2622 00022.

References

/1./ F. N. Hooge, T. G. M. Kleinpenning, and L. K. J. Vandamme, "Experimental studies on 1/f noise," *Reports on Progress in Physics*, vol. 44, no. 5, pp. 479-532, May1981.

/2./ G. P. Zhigal'skii, A. V. Karev, *J. Communications Technology and Electronics* 44 (1999) 206-210

/3./ L. K. J. Vandamme, "On the Calculation of 1/f Noise of Contacts," *Applied Physics*, vol. 11 pp. 89-96, 1976

/4./ L. K. J. Vandamme, "Noise as a Diagnostic Tool for Quality and Reliability of Electronic Devices," *IEEE Transactions on Electron Devices*, vol. 41, no. 11, pp. 2176-2187, Nov.1994.

/5./ V. Sedlakova et al., Current Density Distribution, Noise and Non-linearity of Thick Film Resistors, *CARTS US Scotsdalle March 31, 2003.*

/6./ 14. Y.Awano, M. Shima, Y. Sakuma, Y. Sugiyama, N. Yokoyama and M. Tacano, Temperature controlled RTS noise from a single InGaAs quantum dot, Proc. of the Int. Conf. "Noise in Physical Systems and 1/f Fluctuations", Gainesville, FL, USA (2001) p.359

/7./ M. Schulz and A. Pappas, Telegraph noise of individual defects in the MOS interface, Proc. of the Int. Conf. "Noise in Physical Systems and 1/f Fluctuations", Kyoto, p. 265 (1991)

J. Sikula, V. Sedlakova, P. Dobis
Brno University of Technology, Technicka 8,
616 00 Brno, Czech Republic
Tel/Fax +4205 4114 3398,
E-mail: sikula@feec.vutbr.cz

Prispelo (Arrived): 15.09.2003   Sprejeto (Accepted): 03.10.2003

REACTIVE PLASMA TECHNOLOGIES IN ELECTRONIC INDUSTRY

Miran Mozetič

Plasma laboratory, Institute of Surface Engineering and Optoelectronics, Ljubljana, Slovenia

INVITED PAPER
MIDEM 2003 CONFERENCE
01. 10. 03 - 03. 10. 03, Grad Ptuj

Abstract: Application of novel materials, higher standards and demands for environment protection have lead to introduction of novel advanced methods for materials processing. Most of them are dry and run under heavily non-equilibrium conditions. A mixture of gas is transformed into the state of plasma. Molecules are excited, dissociated and ionized and the resultant radicals are used to treat electronic components. Examples of technologies include chemical plasma cleaning, plasma activation, selective plasma etching and plasma ashing. Chemical plasma cleaning has been introduced in microelectronics as an alternative to physical plasma cleaning, whose drawback is a re-deposition of sputtered material. In classical electronic industry, the chemical plasma cleaning has been introduced as an environmentally benign alternative to wet chemical cleaning. Plasma activation has become the most efficient method for a dramatic increase of the polymers wettability and thus affinity to metallization. Selective plasma etching is an excellent method for treatment of polymer-matrix composites prior to metallization, while plasma ashing is a simple and reliable method for organic dust removal.

Some examples of industrial application of above technologies will be presented. Advantages as well as drawbacks and limitations of the technologies will be discussed.

Reaktivne plazemske tehnologije v elektronski industriji

Izvleček: Uporaba novih materialov, višjih standardov in zahtev po zaščiti okolja so vodili k vpeljavi naprednih postopkov obdelave materialov. Večina sodobnih postopkov je suhih in potekajo pri termodinamsko močno neravnovesnih razmerah. Mešanico plinov pretvorimo v stanje plazme. Plinske molekule se v plazmi vzbudijo, disociirajo in ionizirajo, tako nastale radikale pa uporabimo za obdelavo elektronskih komponent. Primeri tehnoloških postopkov, ki temeljijo na uporabi reaktivne plazme, so kemijsko plazemsko čiščenje, plazemska aktivacija, selektivno plazemsko jedkanje in plazemsko upepeljevanje. Kemijsko plazemsko čiščenje so najprej uporabili v mikroelektroniki kot alternativo fizikalnemu plazemskemu čiščenju, katerega pomanjkljivost je nalaganje razpršenega materiala na površine. V klasični elektronski industriji se kemijsko plazemsko čiščenje počasi uveljavlja kot okolju prijazen nadomestek mokrega kemijskega čiščenja. Plazemska aktivacija je postala najuspešnejša metoda za povečanje omočljivosti polimerov, ki je potrebna za učinkovito metalizacijo plastičnih komponent. Selektivno plazemsko jedkanje je odlična metoda za obdelavo kompozitnih materialov s polimerno matriko, medtem ko je plazemsko upepeljevanje preprosta in zanesljiva metoda za odstranjevanje organskih prašnih delcev z različnih površin. V prispevku je opisana uporaba nekaterih plazemskih postopkov za industrijsko uporabo in prednosti, pomanjkljivosti in omejitve navedenih tehnologij.

1. Introduction

The very traditional trend in electronic industry is miniaturization. Since the beginning of a massive production of microelectronic devices the miniaturization demands required application of novel, by that times unknown technologies. Wet chemical etching has been replaced with advanced dry etching processes such as ion beam etching, reactive ion etching and plasma stripping. Wet chemical deposition methods as well as thermal evaporation have been replaced by a variety of plasma- or ion beam- assisted deposition methods currently often referred as physical (PVD) or chemical (CVD) vapor deposition techniques. Traditional cleaning methods have been replaced with novel plasma based cleaning techniques. More recently, further miniaturization is expected in three-dimensional devices based on novel nano-scale materials.

Classical electronic industry is also undergoing miniatuarization trends. Unlike microelectronic industry where the materials expenses plays a minor role, a driving force of miniaturization of classical electronic devices is the cost of materials. Currently, the classical materials such as metals, polymers and ceramics are being gradually replaced with composites. Promising substitutes for metals are graphite-polymer composites and glassy carbon, while ceramics are replaced with ceramic composites. A major advantage of a carbon-based composite over a metal is a low weight, good mechanical properties and excellent chemical resistance.

A major consideration in both classical electronics and microelectronics is environment. The environment protection laws became severe and the general expectation is even tightening the restrictions. Nowadays any technological effort takes into account the ecological suitability of the materials processing.

2. Reactive plasma

The problems of miniaturization, application of novel materials and environment protection have been solved with application of technologies that are based on application of low-pressure plasmas. Namely, traditional methods were found inadequate to solve novel demands. Traditional methods are based on thermodynamically equilibrium processing techniques. Wet chemical treatments as well as thermal deposition techniques are good examples. The driving force in all these techniques is temperature. The reactivity of particles involved in these techniques depends on temperature. Since many advanced materials do not stand high temperature processing the application of traditional techniques is obviously limited. The only solution to current demands is therefore application of a heavily non-equilibrium media for materials processing.

One of the best examples of non-equilibrium medium is low-pressure plasma. Plasma is sometimes referred as the fourth state of the matter - solid, liquid and gaseous being the first three. It is created in a gas under low pressure when electric current is allowed to pass through. This can be achieved in a variety of gaseous discharges including the DC, RF and MW discharge. In any case, the free electrons are accelerated in an electric field. When they gain enough energy (above the ionization threshold) the electrons can ionize neutral molecules. As more electrons are generated at ionization collisions the number of electrons in the gas increases and finally reach such a large value that the gas becomes conductive and the discharge self-sustained. A simultaneous effect of electron multiplication is a formation of positive ions. The density of positive ions in plasma is often equal to the electron density.

Energetic electrons in plasma are not only capable of ionizing molecules, but they can also suffer other types of inelastic collisions with neutral molecules. The diatomic gaseous molecules can be dissociated, excited in a variety of states including rotational, vibrational and electronic states, and the atoms can be excited in electronic states as well. Plasma therefore consist a variety of particles that are present in a normal gas only in low quantities, such as neutral atoms and highly excited molecules. Since the radiative life time of many states in long (some states are metastable) the particles may be found in extremely high states - in nitrogen plasma, for instance, the average vibrational state is easily about 20 corresponding to the internal vibrational temperature well over 10.000°C. Similarly, the neutral atom density may be also high and the dissociation degree may approach unity, otherwise typical for temperature of 50.000°C.

On the other hand, the kinetic temperature (i.e. average kinetic energy) of gaseous particles often remains close to room temperature. The discrepancy between the internal and the kinetic gas temperature is due to a poor kinetic interaction of energetic electrons with heavy particles (molecules and atoms). At an elastic collision only a fraction (often less than 0.001) of an electron kinetic energy can be transferred to a heavy particle. The kinetic temperature of heavy particles is therefore not much influenced by the electron energy. As long as one can avoid direct heating of heavy particles in electric field and some other effects including superelastic collisions between vibrationally excited molecules and atoms, the kinetic temperature of heavy particles remain low. These conditions, i.e. a high concentration of excited particles at low kinetic temperature, are ideal for advanced treatments of materials in electronic industry. Several technologies based on application of such plasmas have been developed and some are described below.

3. Discharge cleaning

Discharge cleaning (often called plasma cleaning) has been first introduced in microelectronics where the demands of materials cleanliness were the highest. Later, it has been introduced to other industrial branches, and the main reason was a requirement of ecological friendly processing. Discharge cleaning is an ecological benign substitute of wet chemical cleaning. The wet chemical cleaning produces large quantities of used chemicals that pollute environment. Plasma cleaning, on the other hand, produces little, if any, pollutants. Organic impurities are often removed with oxygen plasma, while oxidizing impurities (O, Cl, S) are effectively removed with hydrogen plasma.

Hydrogen plasma has been successfully applied for cleaning silver contacts in switching devices. The major reason for introducing plasma cleaning was ecological - contacts were previously cleaned with freon that has been forbidden due to harmful effects to ozone layer in the upper atmosphere. Figure 1(a) is a typical AES depth profile of a contact as received from a production line. There are several impurities on the surface including organic impurities as well as oxygen, sulfur, chlorine and potassium. The contaminants are effectively removed by hydrogen plasma treatment. A typical treatment time is 1 minute thus suitable for massive industrial production. The AES depth profile after plasma treatment is shown in Figure 1(b). There are hardly any impurities on the surface except of traces of O and S probably due to secondary pollution since samples were exposed to air prior to AES analyses. The effect of perfect surface cleanliness is demonstrated in Figure 1(c) that is a plot of contact resistance of the switching device after plasma cleaning. The contact resistance remains extremely low even at poor contact force.

Copper components heavily contaminated with organic impurities are best cleaned with oxygen plasma followed by a hydrogen plasma treatment. Namely, hydrogen plasma is not most efficient in removing organic compounds. Oxygen plasma treatment is more efficient, but a drawback of oxygen plasma treatment is often a formation of a thin oxide film on the material surface. This oxide film is effectively reduced with hydrogen plasma. Figure 2 demonstrates the efficiency of combined oxygen/hydrogen plasma cleaning. The sample as received from a production line is covered with a variety of impurities, organic compounds being predominated. Chemical treatment reduces amount of impurities substantially but not all. Oxygen plasma treatment effectively removes organic compounds but oxidizes the material. Finally, hydrogen plasma treatment reduces the oxide film leaving the surface virtually atomically clean.

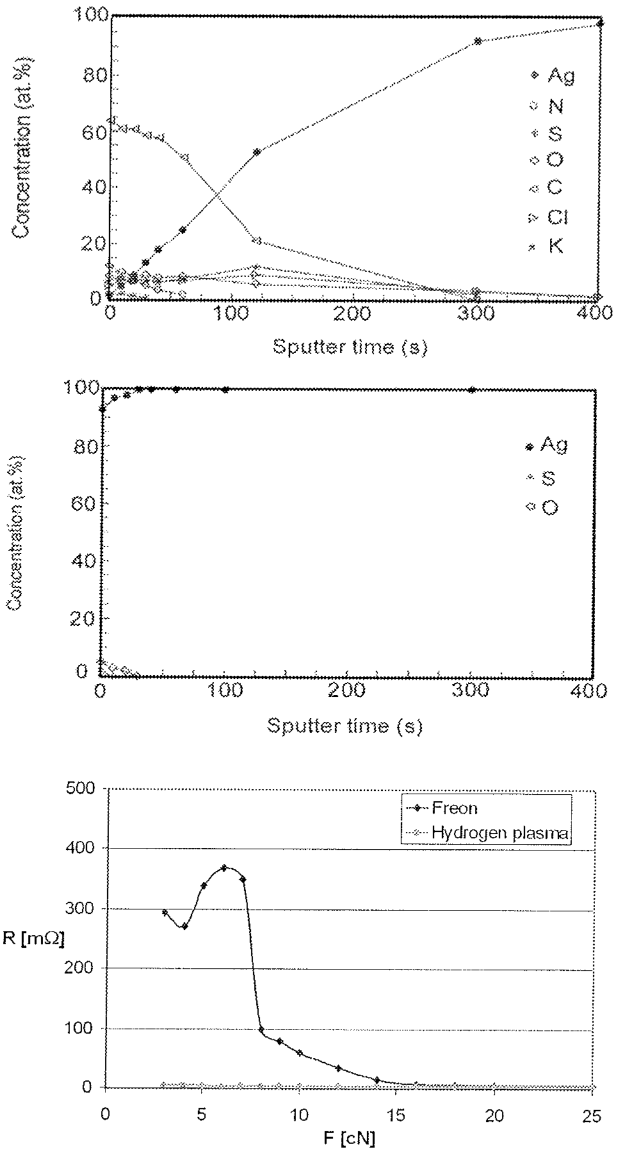
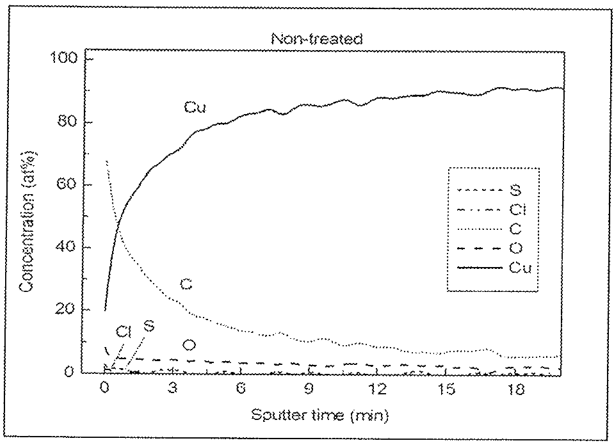


Figure 1. *AES depth profile of a silver contact from a switching device as received (a) and after hydrogen plasma cleaning (b). The contact resistance between the contacts of a plasma cleaned switch compared to conventional freon-cleaned switch (c).*



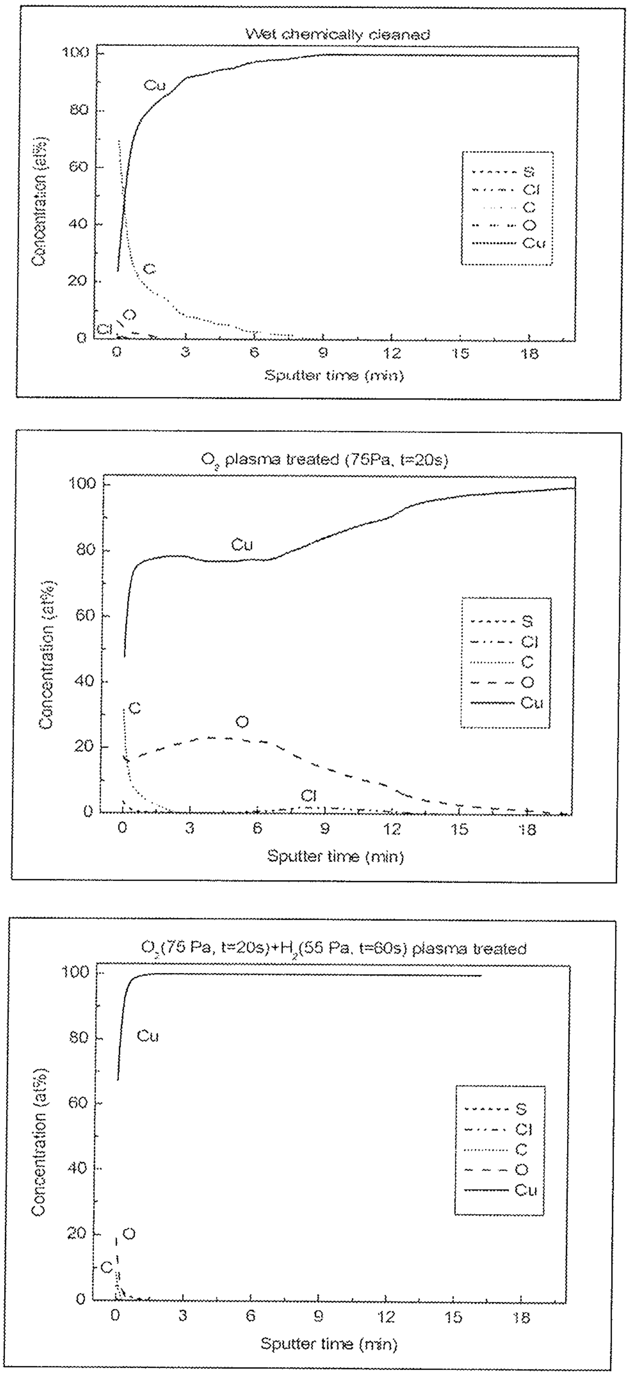


Figure 2. *AES depth profiles of a copper component: as received from production line, wet chemically cleaned, oxygen plasma cleaned and hydrogen plasma cleaned.*

4. Surface activation

Polymers are often made from unpolar groups resulting in a poor surface wettability. In order to increase the wettability prior to painting, printing or metallization, the surface should be activated. Surface activation is performed by a variety of techniques including the wet chemical treatment, UV irradiation, ion beam treatment, laser modification and mechanical roughening, but the best method proved to be low-pressure oxygen plasma oxidation. By plasma treatment the surface become rich with polar groups enhancing the surface wettability dramatically. Figure 3 represents a water drop on a housing of a capacitor prior and after plasma activation. The difference in surface wettability is clear. Even more important is the fact that optimal surface wettability is obtained after less that 0.1 s of plasma treatment - a fact that makes plasma activation extremely suitable for a massive industrial production. Figure 4 presents a contact angle of a water drop on a polymer foil. As expected the contact angle decreases dramatically in first few seconds of plasma treatment, but tends to increase slowly with further plasma action indicating the overestimation of plasma treatment should be avoided.

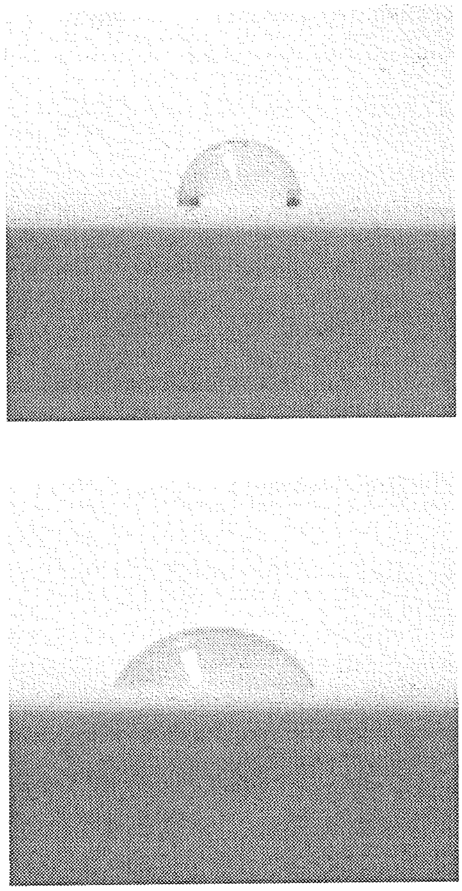


Figure 3. A water drop on a capacitor housing before (left) and after (right) plasma treatment

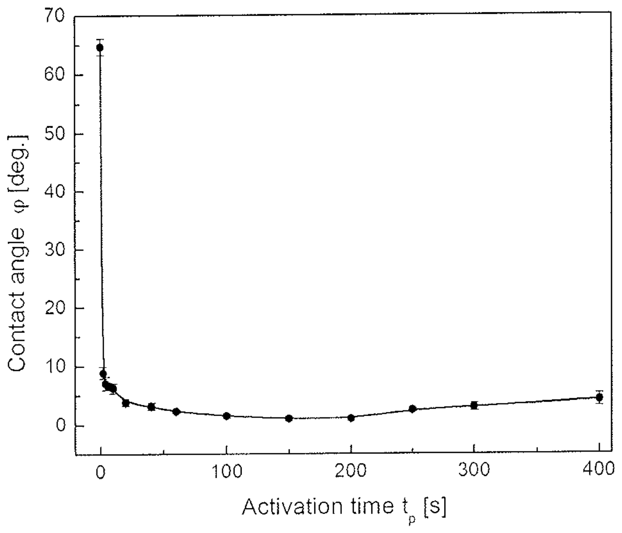


Figure 4. A contact angle of a water drop on a polymer foil

5. Selective plasma etching

Metals are being gradually replaced with polymer matrix composites. The major advantage of a graphite-polymer composite over a metal is a low weight, good mechanical properties and excellent chemical resistance. A major drawback, on the other side, is a poor affinity to metallization. A traditional method of polymer-matrix composite metallization is surface activation with palladium followed with a chemical metallization. An excellent substitute for those ecological unsuitable technologies is plasma treatment. Unlike the upper methods that produce environmentally harmful residues, plasma treatment is an ecologically benign technology. The plasma treatment effects are threefold: first, increases the surface wettability (Figure 5), second, it reducer the concentration of polymer on the surface (Figure 6), and third, it makes the surface rough enough to assure good adhesion between the substrate and the metallization (Figure 7,8).

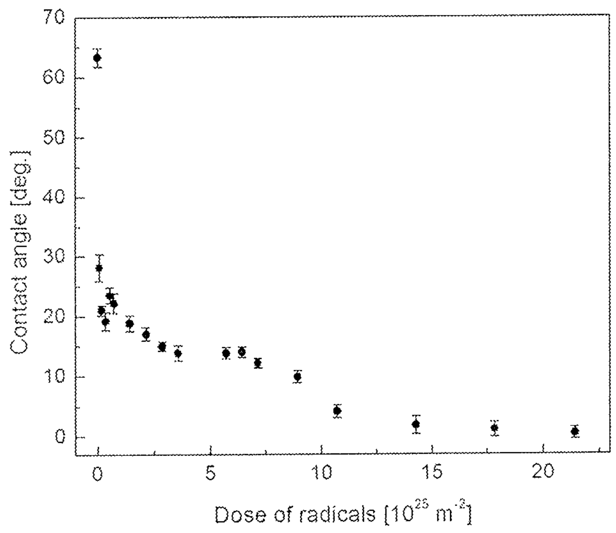


Figure 5. Contact angle of a water drop on a graphite-polymer surface versus the dose of oxygen radicals

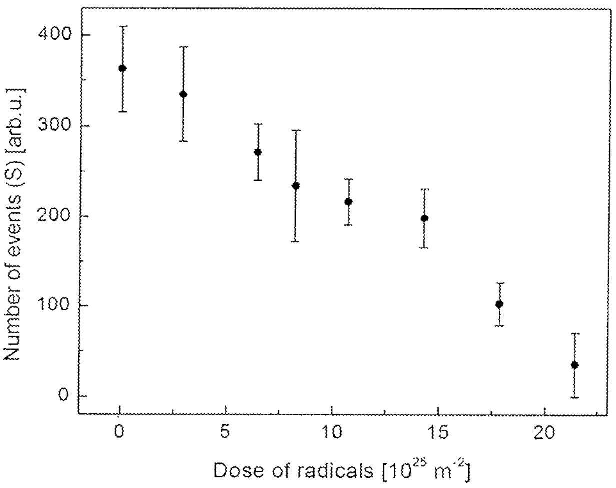


Figure 6. Concentration of sulfur in the surface layer of a graphite-pps polymer composite versus the dose of oxygen radicals.

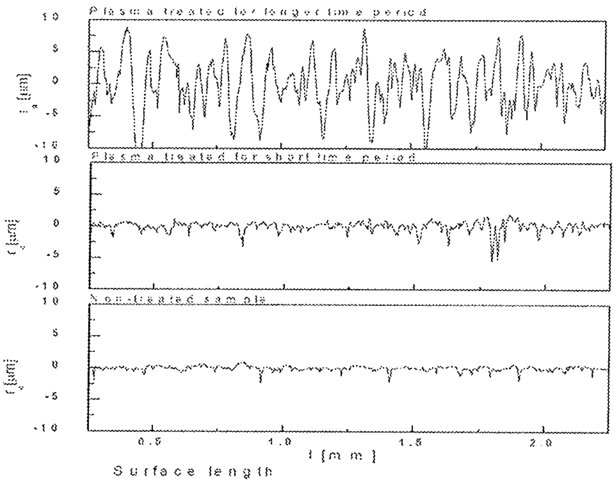


Figure 7. Evolution of a surface roughness of a graphite-pps polymer composite during plasma treatment.

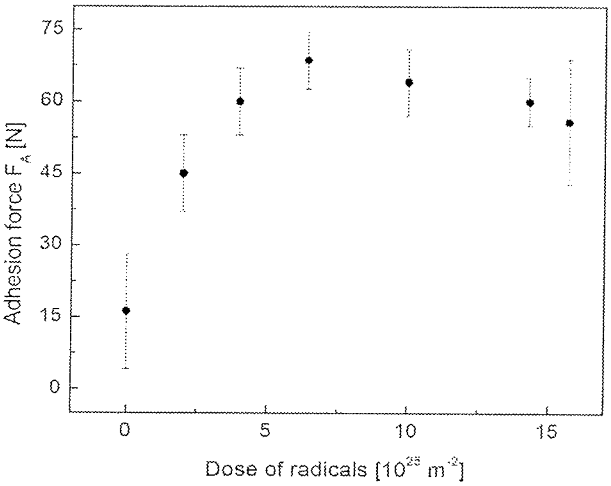


Figure 8. Adhesion force of a metallization layer on a graphite polymer-matrix composite.

The selective plasma etching technology is also an excellent method for treatment of a variety of polymer matrix composites to reveal the distribution and orientation of different particles in the matrix. It is the unique method to determine the composition of different films and paints. Figure 9(a,b,c) represent scanning electron images of a photographic film during plasma etching. Untreated samples reveal no significant structure. A 30s plasma treatment reveals spherical holes indicating the presence of gaseous bubbles in the uppermost layer of a photographic film. The grains of silver halide are observed only after prolonged plasma treatment.

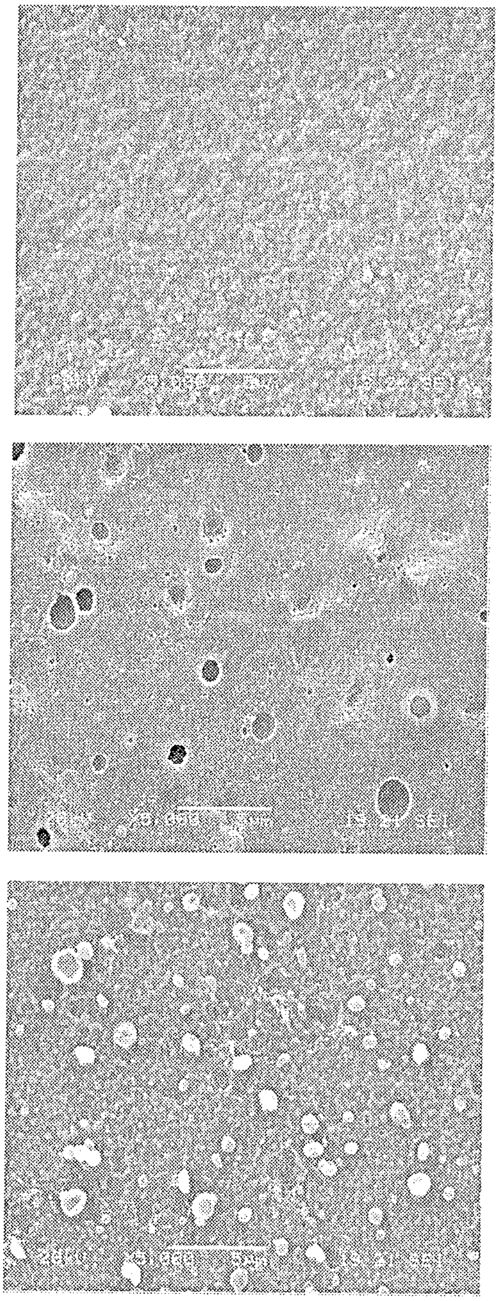


Figure 9. SEM images of a photographic film before plasma treatment (upper), after a short plasma treatment (middle) and prolonged (lower) treatment

Another example of application of the selective plasma technology is an advanced paint for automotive industry. Figure 10 represents a SEM image of the paint before and after plasma treatment. It is clear that plasma treatment clearly reveals the original distribution of mica flakes in the coating.

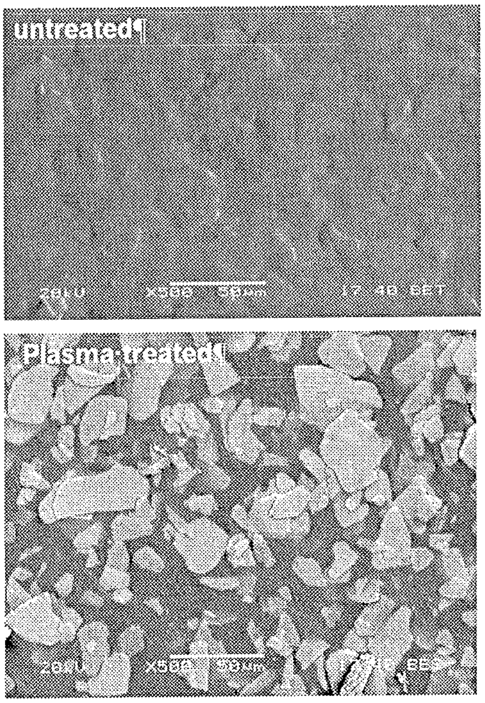


Figure 10. A SEM image of a mica paint before and after oxygen plasma treatment

6. Conclusions

In the past few years the reactive plasma technologies have been successfully applied to modern electronic industry. Apart from the technical advantages, the major reason for introduction plasma based technology is environment protection. The reactive plasma technologies are usually ecological benign alternatives to wet chemical treatments. The maintenance as well as consumables costs are often much lower than corresponding costs of traditional techniques. The major drawback of novel technologies, however, is a high cost of plasma reactors. It is expected that these expenses will decrease in the next future, as more users of reactors will appear in the market.

References

/1/ B. Chapman, Glow Discharge Processes (Willey, New York, 1980).

/2/ H. Boeing, Plasma Science and technology (Cornell Univerity Press, London, 1982).

/3/ M. R. Wertheimer, L. Martinu and E. M. Liston, Plasma sources for polymer surface treatment, Handbook of thin film Process Technology, ed. by D.A.Glocker and S.I. Shah. Bristol (Inst. Of Physics Publishing, Bristol, 1998).

/4/ A. Ricard , Reactive plasmas (Societe Francaise du Vide, Paris, 1996).

/5/ I. Šorli, W. Pettasch, B. Kegel, H. Schmidt and G. Liebel, Inform. Midem 26, 35 (1996).

/6/ I. Sorli and R. J. Rocak, Vac. Sci. Technol. A18, 338 (2000).

/7/ Babič, D., Poberaj, I., Mozetič, M., Rev. Sci. Instr. 72, 4110 (2001).

/8/ Poberaj, I., Babič, D., Mozetič, M., J. Vac. Sci. Technol. A20, 189 (2002).

/9/ Mozetic, M., Ricard, A., Babic, D., Poberaj, I., Levaton, J., Monna, V., Cvelbar, U., J. Vac. Sci. Technol. A21, 369 (2003).

/10/ Wise, H., Wood, B.J., Gas and Surface Reaction Collisions, Advances in Atomic and Molecular Chemistry, ed. D.R. Bates, I. Estermann, Academic Press, New York (1967).

/11/ M. Mozetič and A. Zalar, Appl. Surf. Sci. 158, 263 (2000).

/12/ P. Pelicon, M. Klanjšek-Gunde, M. Kunaver, J. Simčič and M. Budnar, Nucl. Instrum. Meth. 190, 370 (2002).

/13/ M. Klanjšek-Gunde and M. Kunaver, Appl. Spectrosc. 57, 1266 (2003).

/14/ M. Kunaver, M. Klanjšek - Gunde, M. Mozetič and A. Horvat, Surf. Coat. Inter. B 86, 175 (2003).

/15/ S. Gomez, P. G. Steen and W. G. Graham, Appl. Phys. Lett 81, 19 (2002).

/16/ H. Singh, J. W. Coburn and D. B. Grawes, J. Appl. Phys 88, 3748 (2000).

/17/ D. J. Wilson, N. P. Rhodes and R. L. Williams, Biomaterials 24, 5069 (2003).

/18/ Z. Y. Wu, N. Xanthopoulos, F. Reymond, J. S. Rossier and H. H. Girault, Electrophoresis 23, 782 (2002).

Miran Mozetič
Plasma laboratory,
Institute of Surface Engineering and Optoelectronics,
Teslova 30, 1000 Ljubljana, Slovenia

Prispelo (Arrived): 15.09.2003 Sprejeto (Accepted): 03.10.2003

DATA-STREAM-BASED COMPUTING: MODELS AND ARCHITECTURAL RESOURCES

Reiner Hartenstein

Kaiserslautern University of Technology, Germany

INVITED PAPER
MIDEM 2003 CONFERENCE
01. 10. 03 - 03. 10. 03, Grad Ptuj

Abstract: The paper addresses a broad readership in information technology, computer science and related areas, introducing reconfigurable computing, and its impact on classical computer science. It points out trends driven by the mind set of data-stream-based computing.

Računanje na osnovi podatkovnih tokov: modeli in arhitekturna sredstva

Izvleček: Prispevek naslavlja bralce s področja informacijske tehnologije, računalništva in sorodnih področij z uvedbo pojma rekonfiguracijsko računanje in njegovega vpliva na klasično računalništvo. Poudarja predvsem trende, katerih gonilo je računanje na osnovi podatkovnih tokov.

1. Introduction

An alternative general purpose platform. The dominance of the instruction-stream-based procedural mind set in computer science stems from the general purpose properties of the ubiquitous von Neumann (vN) microprocessor. Because of its RAM-based flexibility no costly application-specific silicon is needed. Throughput is the only limitation by its sequential nature of operation (von Neumann bottleneck). Now a second RAM-based computing paradigm is heading for mainstream: *morphware*, electrically reprogrammable by reconfiguration of its structure /1/. This is a challenge to CS curricula innovators, also an occasion to reconsider criticism of the von Neumann culture /2/ /3/ /4/ /5/.

CS to explore new horizons. From this starting point Computing Sciences (CS) are slowly taking off to explore new horizons: a dichotomy of two basic computing paradigms, removing the blinders from the still dominant von-Neumann-only mind set, which is still ignoring the impact of Reconfigurable Computing (RC). It has been predicted, that by the year 2010 more than 90% of all programmers will implement applications for embedded systems, where a procedural / structural double approach is a pre-requisite. Currently programmers do not yet have the background required for this new labor market. This challenge can be met only by the dichotomy of machine paradigms within CS.

The education gap can be bridged. A rich supply of tools and research results is available to adapt fundamental courses, lab courses and exercises /6/. There are a lot of similarities between both branches, like between matter and anti matter. But also some challenges are waiting. Our basic curricula do not teach, that hardware and software are alternatives, and, how hardware / software partitioning is carried out. E. g. some urgently needed new directions of algorithmic cleverness are not yet taught. For instance, how to implement a high performance application for low power dissipation on 100 processors running at 200 MHz, rather than on one processor running at 20 GHz. A curricular revision is overdue /7/.

2. Reconfigurable computing

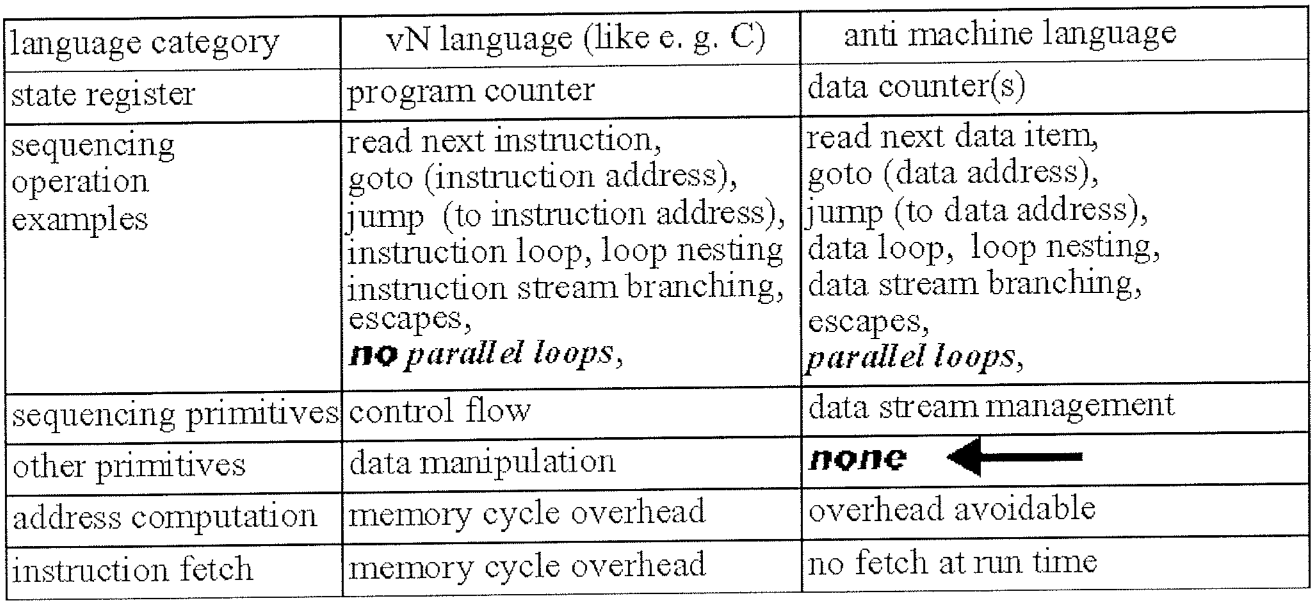
> In morphware application the lack of algorithmic cleverness is an urgent educational problem.

Advancing maturity is indicated by a growing consensus on terminology (fig. 1). Occupied by other areas. the term "dataflow machine" /8/ and the acronym DSP should not be used. So this paper uses the term *anti machine*.

platform category	source "running" on platform	machine paradigm
hardware	(hardwired)	
morphware	configware	
ISP*	software	von Neumann
AM*	flowware	anti machine
rAM*	flowware & configware	

*) acronyms see fig. 4, terminology: fig. 8 and 9.

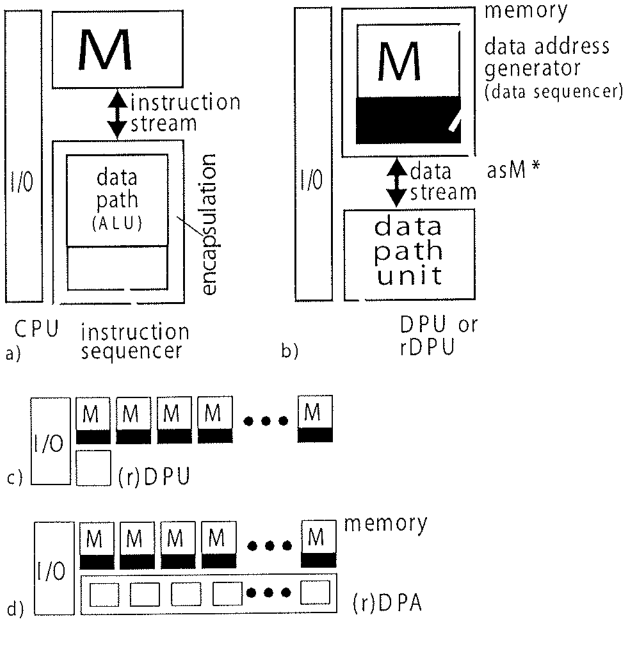
Fig. 1: Platform categories



language category	vN language (like e. g. C)	anti machine language
state register	program counter	data counter(s)
sequencing operation examples	read next instruction, goto (instruction address), jump (to instruction address), instruction loop, loop nesting instruction stream branching, escapes, ***no parallel loops***,	read next data item, goto (data address), jump (to data address), data loop, loop nesting, data stream branching, escapes, ***parallel loops***,
sequencing primitives	control flow	data stream management
other primitives	data manipulation	**none** ⟵
address computation	memory cycle overhead	overhead avoidable
instruction fetch	memory cycle overhead	no fetch at run time

Fig. 2: Traditional Software languages versus Flowware languages.

The dichotomy of fundamental models. More important is the terminology from a global point of view (figure 1 a). Whereas classical CS deals with *software* (SW) running on *hardware (HW)*, the new branch deals with *flowware (FW)* /9/ running on *HW*, or, *configware (CW)* /10/ and *FW* "running" on *morphware (MW)* /11/,. This paper gives introductions for a broad readership mainly with a CS background..



*) auto-sequencing memory

Fig. 3: Illustration of basic machine paradigms: a) von Neumann, b) data-streambased anti machine with simple DPU, c) with rDPU and distributed memory architecture, d) w. DPU array (DPA or rDPA).

This paper does not deal with fine grain morphware (FPGAs, using single bit wide CLBs) already being mainstream. *Reconfigurable Computing* (RC) uses coarse grain morphware platforms: rDPUs (reconfigurable data path units), which, similar to ALUs, have major path widths, like 32 bits, for instance - or even rDPAs (rDPU arrays). Important applications are derived from the decay of "general purpose" vN computer architecture /2/ /3/ /4/ and its performance limits /5/, creating a demand for accelerators. For very high throughput requirements RC is the drastically more powerful and more area-efficient and energy-efficient programmable alternative /5/ /12/ to FPGAs (fig. 6), also providing a massive reduction of configuration memory and time needed for configuration /13/.

AM	anti machine (DS machine)
asM	autosequencing Memory
rAM r	econfigurable AM
CPU	"central" processing unit: DPU **and** instruction sequencer (vN)
CS	Computing Sciences, Computer Science
CW	configware
DPU	data path unit **without** sequencer
rDPU	reconfigurable DPU
DPA	data path array (DPU array)
rDPA	reconfigurable DPA
DS	data stream
DSM	data stream processing machine
EE	Electrical Engineering
ESW	embedded SW
FW	flowware
HW	hardware
ISP	instruction stream processor
MW	morphware
RC	reconfigurable computing
SW	software
vN	von Neumann (machine paradigm)

Fig. 4: Some acronyms.

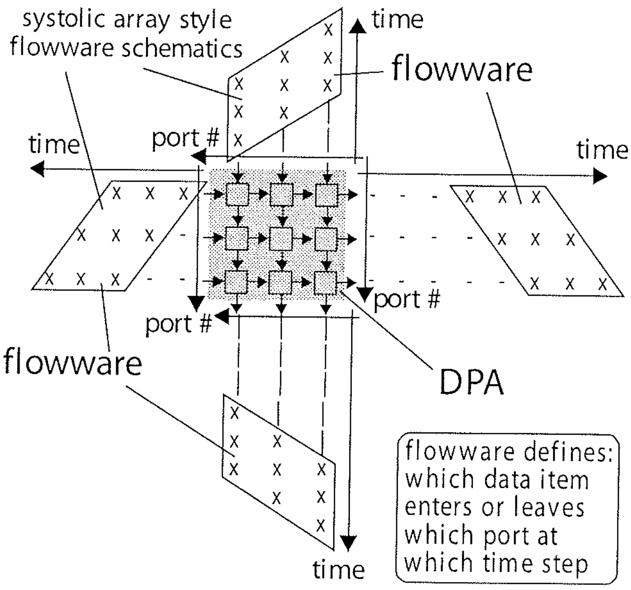


Fig. 5: Flowware.

Commercial architectures. In application areas like multimedia, wireless telecommunication, data communication and many others, the throughput requirements are growing faster than Moore's law, along with growing flexibility requirements due to unstable standards and multi-standard operation /14/. Currently the requirements can be met from commercial sources only by rDPAs from a provider like PACT /15/ /16/ /17/ /18/ /19/ (fig. 11).

Domain-specific approach. A a currently viable solution appears the domain-specific approach /13/, where a design space explorer may help to derive within a short time an optimum (r)DPU and (r)DPA architecture from a benchmark or domain-typical set of applications /20/ /21/.

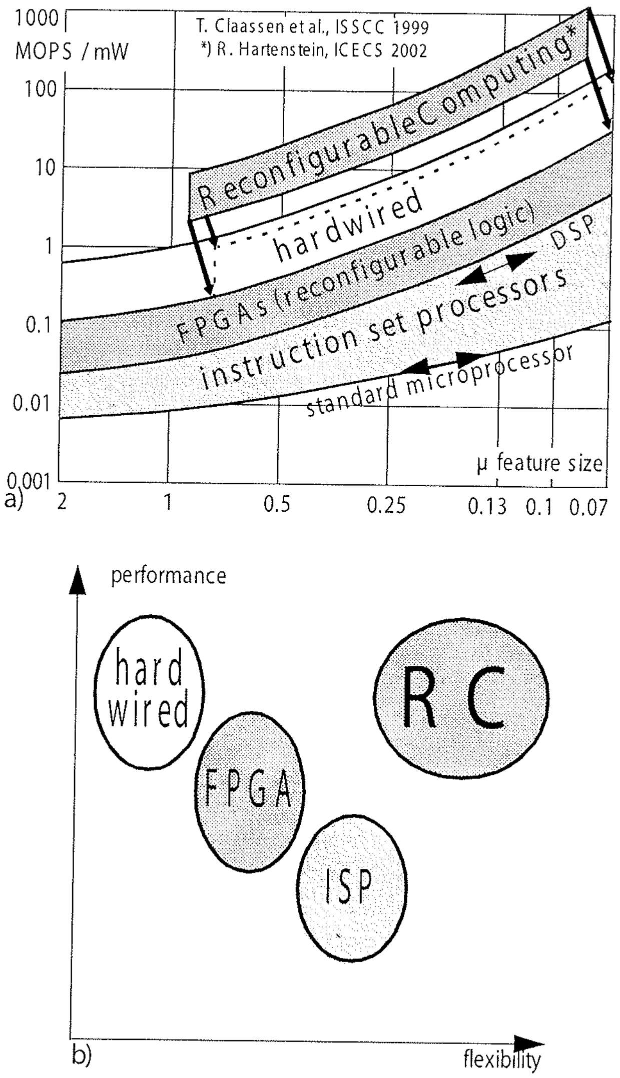


Fig. 6: Energy efficiency and performance vs. flexibility incl. Reconfigurable Computing.

3. Data-stream-based computing

Traditional instruction-stream-based informatics is based on computing in the time domain, where a program deserves scheduling the instructions for execution (fig. 9). Classical basic structures and principles in computing are

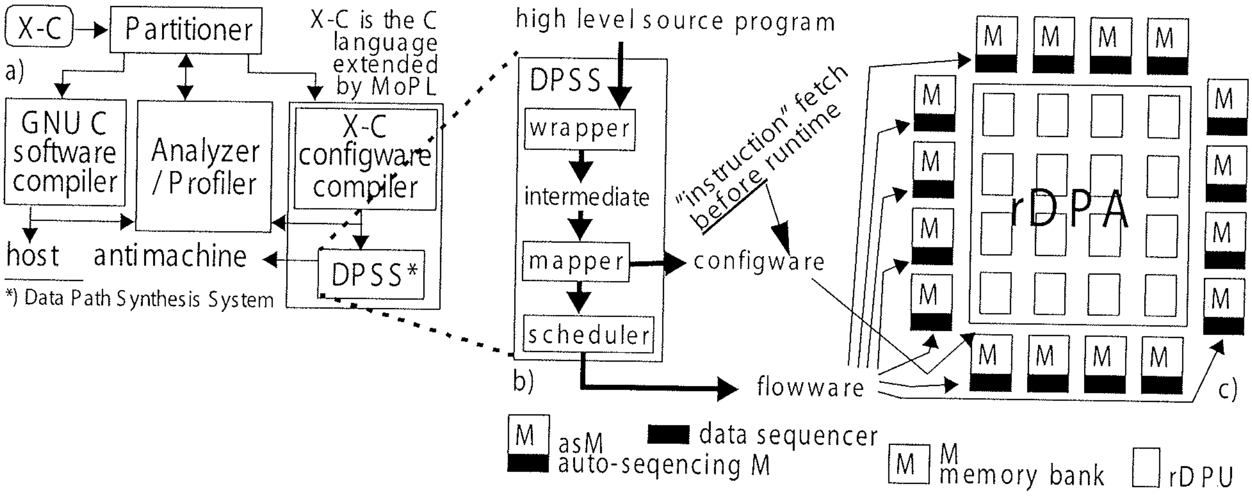


Fig. 7: CW/ SW Co-Compilation: a) CoDe-X partitioning Co-Compiler, b) DPSS details, c) anti machine target.

von-Neumann-centric, which are instruction-stream-based, where instruction sequencer and datapath are in the same CPU (fig. 3 a). Due to reconfigurable a second basic model has emerged, so that we now have a dichotomy of models: instruction-stream-based computing vs. data-stream-based computing. There is a lot of similarities, so that each of the 2 models is a kind of mirror image of the other model - like with matter and antimatter.

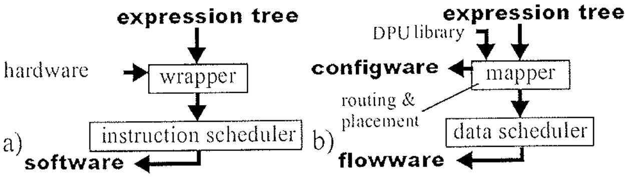


Fig. 8: Compilation: a) von-Neumann-based, b) for anti machines

Data counters replace the program counter. Datastream-based computing, the counterpart of instruction-stream-based von Neumann computing (fig. 9), however, uses one or more data counters instead of a single program counter (example in fig. 3 b). However, there are some asymmetries, like predicted by Paul Dirac for antimatter. Figure 7 b shows the block diagram of data-stream machine with 16 autosequencing memory banks. The basic model allows the machine to have 16 data counters, where as a von Neumann machine cannot have more that one program counter. The partitioning scheme of the data-stream machine model assigns a sequencer (address generator) always to a memory bank, never to a DPU. This modelling scheme goes fully conform with the area of embedded distributed memory design and management (see section on Embedded Memory).

The vN microprocessor is indispensable. But because of its monopoly our CS graduates are no more professionals.

Flowware. Data streams have been popularized by systolic arrays /22/ /23/ /24/ (fig. 5), the super systolic array /25/, and more recently by projects like SCCC /26/, SCORE /27/ /28/, ASPRC /29/, BEE /30/ /31/ /32/, the KressArray Xplorer /20/ /21/ and many other projects. In a similar way like instruction streams can be programmed from *SW sources*, also data streams can be programmed, but from *FW sources*. High level programming languages for flowware /33/ and for software join the same language principles and have a lot in common, no matter, wether finally the program counter or a data counter is manipulated. Figure 8 illustrates the basic semantic principles of flowware by 12 data streams associated with the 12 ports of a DPA. The data schedule generated from a flowware source determines, which data object has to enter or leave which DPA port (or DPU port) at which time. This way flowware can be used to program the 12 autosequencing memory banks (asM) of the embedded distributed memory to generate the expected data streams.

machine category	(a) instruction set processor	(b,c) data stream processor	
		(b) hardwired	(c) morphware
machine paradigm	von Neumann (vN)	anti machine	
reconfigurability support	no	no	yes
programming	instruction-procedural	no	structural (super "instruction" fetch)
		data scheduling	
program source	software	flowware	flowware & configware
"instruction" fetch	at run time	at fabrication time	before run time
execution at run time	instruction schedule	data schedule	
operation spin	instruction flow	data stream(s)	
operation resources	CPU	DPU, or, DPA	rDPU, or, rDPA
	hardwired	hardwired	reconfigurable
parallelism	only by multiple machines	by single machine or multiple machines	
state register	single program counter	one or more data counter(s)	
state register located	within CPU	outside DPU or DPA:	outside rDPU or rDPA:
		within asM (autosequencing memory banks)	

Fig. 9: Asymmetry between machine and anti machine paradigms.

Two programming sources. Figure 7 a, Figure 8 a and Figure 10 d illustrate, why a von Neumann machine needs just software as the only programming source, since the resource part being hardwired is not programmable. Figure 7 b, Figure 8 b and Figure 10 e show, why a reconfigurable data-stream-based machine needs two programming sources: configware to program (to reconfigure) the operational resources, and, flowware to schedule the data streams. Figure 10 f shows why hardwired anti machines need only a single program source: flowware only. Figure 7 c illustrates the structure of the compiler (DPSS /25/) generating the code of both sources from a high level programming language source (here a C subset /25/): phase 1 performs routing and placement to configure the rDPA, and phase 2 generates the flowware code to program the autosequencing distributed memory, so that the data streams fit to the routing and placement result from phase 1.

The same model for hardware and morphware. There is in principle no difference, whether a data-stream-based DPAs is hardwired or reconfigurable. The only important difference is binding time of placement and routing: before fabrication, or, after fabrication (compare fig. 9 b).

Embedded Distributed Memory. Together with application-specific embedded memory architecture synthesis also flowware implementation (for memory management strategies) is subject of performance and power optimization /34/, also by loop transformations /35/. Good flowware may be also obtained after optimized mapping an application onto rDPA /20/, where both, data sequencers and the application can be mapped (physically, not conceptually) onto the same rDPA /13/.

Memory bandwidth. To solve the memory communication bandwidth problem the anti machine paradigm (datastream-based computing) is much more efficient than "von Neumann". There are alternative embedded memory implementation methodologies available /34/ /36/ /37/ /38/, either specialized memory architecture using synthesized address generators (e. g. APT by IMEC /34/), or, flexible memory architectures using programmable general purpose address generators /39/ /40/. Performance and power efficiency are supported especially by sequencers, which do not need memory cycles even for complex address computations /34/, having been used also for a smart memory interface of an early anti machine architecture /41/ /42/.

Data-Stream-based vs. concurrent Computing. Classical parallelism by concurrent computing has a number of disadvantages over the parallelism by anti machines having no von Neumann bottleneck, what is discussed elsewhere /32/ /42/. Amdahls law explains just one of several reasons of inefficient resource utilization. vN-type processor chips are almost all memory, because the architecture is wrong. Here the metric for what is a good solution has been wrong all the time.

4. Configware compilers

Co-Compilation. Using coarse grain morphware (rDPAs) as accelerators changes the scenario: implementations onto both, host and accelerator(s) are RAM-based, which allows turn-around times of minutes for the entire system, instead of months for hardwired accelerators, and, supporting a migration of accelerator implementation from IC vendor to customer, who usually does not have hardware experts. This creates /43/ a demand for compilers accepting high level programming language (HLL) sources. Partly dating back to the 70ies and 80ies know-how is available from the classical parallelizing compiler scene, like

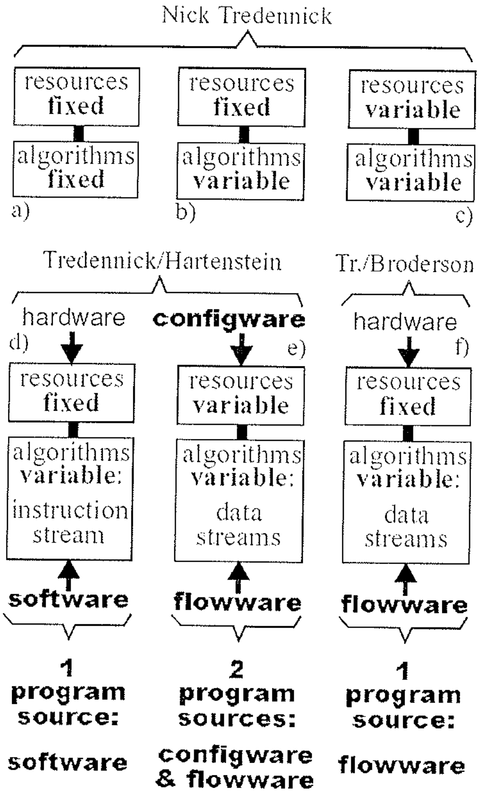
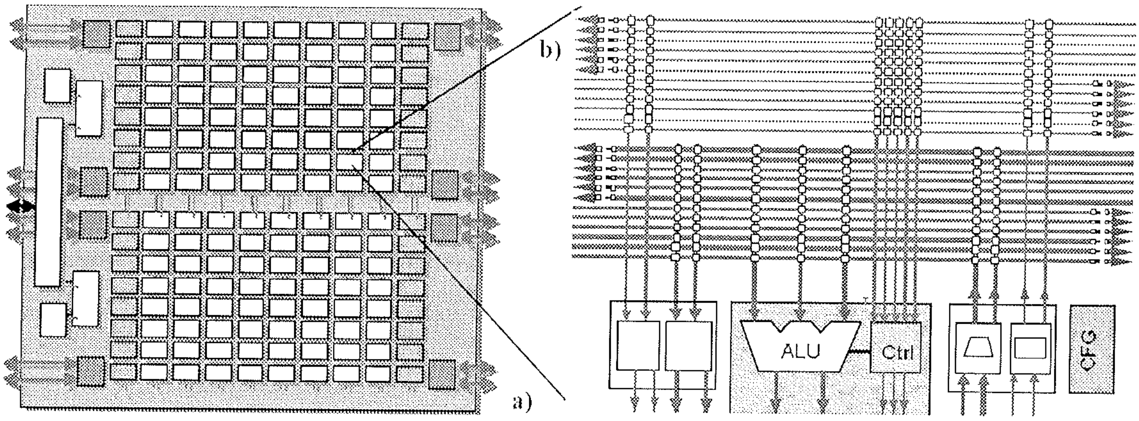


Fig. 10: Nick Tredennick's digital system classification scheme: a) hardwired, b) programmable in time, c) reconfigurable d) von-Neumann-like machine paradigm e) reconfigurable anti machine paradigm f) Broderson's hardwired anti machine. terminology also from: /5/.



c) platform	application example	speed-up factor	method
PACT Xtreme 4-by-4 array [2003]	16 tap FIR filter	x16 MOPS / mW	straight forward
MoM anti machine with DPLA* [1983]	grid-based DRC** 1-metal 1-poly nMOS 256 reference patterns	x2000 (computation time)	multiple aspects

*) Kaiserslautern e-programmable PLA, manufactured by E.I.S. project MPC organization
**) Design Rule Check based on 4-by-4 pixel reference patterns

Fig. 11. Configurable System-on-Chip with XPU (xtreme processing unit) from PACT AG): a) XPU array structure, b) the structure of a rDPU, c) speed up factors (PACT & MoM).

software pipelining /43/, and, loop transformations /44/ /45/ /46/ /47/ (survey in /48/).

Mapping applications onto rDPAs. Classical systolic arrays could be used only for applications with regular data dependencies, because at that time linear projections or algebraic methods had been used for mapping, which yield only uniform arrays with strictly linear pipes. However, today for DPA synthesis or mapping applications onto rDPAs simulated annealing is used instead, to avoid the limitation to regular data dependencies /5/ /25/. This *("super systolic array")* generalization of the systolic array by Kress /49/ also supports inhomogenous irregular arrays, supporting also any wild shapes of pipes within rDPA pipe networks /20/ /21/.

Automatic partitioning. Until recently, not only for hardware / software co-design, but also for software / configware design, the compiler is a more or less isolated tool used for the host only. But accelerators are still implemented by CAD. *Software / configware partitioning is still done manually* /27/ /50/, requiring massive hardware expertise, particularly when hardware description language (HDL) and similar sources are used. Compilation from HLL sources /25/ /26/ /43/ /51/ still stem from academic efforts, as well as the first automatic *cocompilation* from HLL sources including automatic software/configware partitioning /52/ (fig. 7 a) by identifying parallelizable loops /5/ / 35/, having been implemented for the data-streambased MoM (Map-oriented Machine) /21/ /39/ /42/.

4.1 Machine paradigms and other general models

Simplicity of the machine paradigm. Machine paradigms are important models to alleviate CS education and for understanding implementation flows or design flows. The simplicity of the von Neumann paradigm helped a lot to educate zillions of programmers. Figure 3 a shows the simplicity of the block diagram, which has exactly one CPU and exactly one RAM module (memory M). The instruction sequencer and the DPU (datapath unit) are merged to be encapsulated within the CPU (central processing unit), whereas the RAM (memory M) does not include any sequencing mechanism. Other important attributes are the RNI mode (read next instruction) and a branching mechanism for sequential operation (computing in the time domain.) Figure 9 compares both machine paradigms. Since compilers based on the "von Neumann" machine paradigm do not support morphware we need the datastream-based anti machine paradigm (sometimes called Xputer paradigm/ 52/) for the rDPA side, (based on *data sequencer* /53/).

The anti machine has no von Neumann bottleneck.

The *Anti Machine* Paradigm for morphware /42/ /55/ and even for hardwired anti machines the data-streambased anti machine paradigm is the better counterpart (fig. 3 b) of the von Neumann paradigm (fig. 3 a). Instead of a CPU the anti machine has only a DPU (datapath unit) without any sequencer, or a rDPU (reconfigurable DPU) without a sequencer. The anti machine model locates data sequencers on the memory side (fig. 3 b). Anti machines do not have an instruction sequencer. Unlike "von Neumann" the anti machine has no von Neumann bottleneck by allowing multiple data counters (fig. 3 c) to support multiple data streams from/to multiple autosequencing memory banks (fig. 3 c) allowing multi-port operational resources much more powerful than ALU or simple DPU: major DPAs or rDPAs (fig. 3 d).

General purpose anti machine. The anti machine is as universal as the von Neumann machine. The anti programming language is as powerful as von-Neumann-based languages. But instead of a "control flow" sublanguage a "data stream" sublanguage like *MoPL* /33/ recursively defines *data goto, data jumps, data loops, nested data loops,* and *parallel data loops*. For the anti machine paradigm all execution mechanisms are available to run such an anti language. Its address generator methodology includes a variety of escape mechanisms needed to interrupt data streams by decision data or tagged control words inserted in the data streams /55/. Figure 9 compares both paradigms.

Architectural resources, conform with the discipline of embedded distributed memory. The anti machine model, where the DPUs are transport-triggered by arriving data, goes conform with the new and rapidly expanding R&D area of embedded distributed memories /34/ /37/ /37/, including the architectural ressources, like application-specific or programmable data sequencers (see /40/ /53/ /54/).

5. Turning PC into PS (Personal Supercomputer)

Many application areas. There is a number of HPC application areas, where the desired performance is hard to reach by "traditional" high performance computing. For instance, the gravitating n-body-problem is one of the grand challenges of theoretical physics and astrophysics /56/. Also hydrodynamic problems fall in the same category, where often numerical modeling can be used only on the fastest available specialized hardware. Analytical solutions exist only for a limited number of highly simplified cases. For interpretation of dense centers of galactic nuclei observed with the Hubble Space Telescope to unite the hydrodynamic and the gravitational approach within one numerical scheme. Until recently this limited the maximum particle number to about a 105 even on largest supercomputers available. The situation improved by the GRAPE special purpose computer /57/. To improve the flexibility a hybrid solution has been introduced with AHAGRAPE, which includes auxiliary morphware (FPGA-based processors) /58/. Another morphware usage example is cellular wireless communication, where the performance requirements grow faster than Moore's law /59/ /60/.

6. Conclusions

The paper has given an introductory survey on reconfigurable logic and reconfigurable computing, and its impact on classical computer science. It also has pointed out future trends driven by technology progress and innovations in EDA. It has tried to highlight, that deep submicron allows SoC implementation, and the silicon IP business reduces entry barriers for newcomers and turns infrastructures of existing players into liability. The paper tried to illustrate, why many system-level integrated future products without reconfigurability will not be competitive. Instead of technology progress better architectures by reconfigurable platform usage will be the key to keep up the current innovation speed beyond the limits of silicon. The paper advocates that it is time to revisit past results from morphware-related R&D to derive promising commercial solutions, and, that curricular updates in basic CS education are urgently needed. The exponentially increasing of CMOS mask costs demands urgently adaptive and re-usable silicon area, which can be efficiently realized by integrating (dynamically) reconfigurable hardware parts on different granularities into sSoCs with great potential for short time-to-market (-> risk minimization), multipurpose/-standard features incl. comfortable application updates within product life cycles (-> volume increase: cost decrease). This results in the fact that several major industry playersare currently integrating reconfigurable cores/datapaths into their processor architectures and system-on-chip solutions.

7. Literature

/1/ J. Becker, R. Hartenstein (invited paper): Configware and Morphware going Mainstream; Journal on System Architecture (JSA), 2003

/2/ Arvind et al.: A critique of Multiprocessing the von Neumann Style; Proc. ISCA 1983

/3/ G. Bell (keynote): All the Chips Outside: The Architecture Challenge; Proc. ISCA 2000

/4/ J. Hennessy: ISCA25: Looking Backward, Looking Forward; Proc. ISCA 1999

/5/ R. Hartenstein (invited paper): The Microprocessor is no more General Purpose, Proc. ISIS'97

/6/ R. Hartenstein (opening keynote): Are we really ready for the Break-through?; Proc. Reconfigurable Architectures Workshop (RAW 2003), Nice, France, April, 2003

/7/ R. Hartenstein (keynote): A Mead-&-Conway-like Breakthrough is overdue; Dagstuhl, July 2003

/8/ D. Gajski et al.: A second opinion on dataflow machines; Computer, Feb 1982

/9/ : http://xputers.informatik.unikl.de/staff/hartenstein/lot/ICECS2002Hartenstein.ppt

/10/ J. Becker et al.: Parallelization in Co-Compilation for Configurable Accelerators; Proc. ASP-DAC ´98

/11/ coined within the Adaptive Computing Programme, funded by DARPA

/12/ A. DeHon: The Density Advantage of Reconfigurable Computing; Computer, April 2000

/13/ R. Hartenstein (embedded tutorial): A Decade of Research on Reconfigurable Architectures - a Visionary Retrospective; DATE 2001, Munich, March 2001

/14/ J. Becker, T. Pionteck, M. Glesner: An Application-tailored Dynamically Reconfigurable Hardware Architecture for Digital Baseband Processing; SBCCI 2000

/15/ http://pactcorp.com

/16/ V. Baumgarten, et al.: PACT XPP - A Self-Reconfigurable Data Processing Architecture; ERSA 2001

/17/ J. Becker, A. Thomas, M. Vorbach, G. Ehlers: Dynamically Reconfigurable Systems-on-Chip: A Core-based Industrial/Academic SoC Synthesis Project; IEEE Workshop Heterogeneous Reconfigurable SoC; April 2002, Hamburg, Germany

/18/ J. Cardoso, M. Weinhardt: From C Programs to the Configure-Execute Model; DATE 2003

/19/ M. Vorbach, J. Becker: Reconfigurable Processor Architectures for Mobile Phones; Reconfigurable Architectures Workshop (RAW 2003), Nice, France, April, 2003

/20/ U. Nageldinger et al.: KressArray Xplorer: A New CAD Environment to Optimize Reconfigurable Datapath Array Architectures; Proc. ASP-DAC 2000.

/21/ U. Nageldinger et al.: Generation of Design Suggestions for Coarse-Grain Reconfigurable Architectures; Proc. FPL 2000

/22/ J. McCanny et al. (Editors): Systolic Array Processors; Prentice Hall; 1989

/23/ M. Foster, H. Kung: Design of Special-Purpose VLSI Chips: Example and Opinions. ISCA 1980

/24/ H. T. Kung: Why Systolic Architectures? IEEE Computer 15(1): 37-46 (1982)

/25/ R. Kress et al.: A Datapath Synthesis System for the Reconfigurable Datapath Architecture; ASP-DAC'95

/26/ J. Frigo, et al.: Evaluation of the streams-C C-to-FPGA compiler: an applications perspective; FPGA 2001

/27/ T. J. Callahan: Instruction-Level Parallelism for Reconfigurable Computing; FPL'98

/28/ E. Caspi, et al.: Extended version of: Stream Computations Organized for Reconfigurable Execution (SCORE): Proc. FPL '2000

/29/ T. Callahan: Adapting Software Pipelining for Reconfigurable Computing; CASES 2000

/30/ C. Chang, K. Kuusilinna, R. Broderson, J. Rabaey: The Biggascale Emulation Engine; summer retreat 2001, UC Berkeley

/31/ H. Kwok-Hay So, BEE: A Reconfigurable Emulation Engine for Digital Signal Processing Hardware; M.S. thesis, UC Berkeley, 2000

/32/ C. Chang, K. Kuusilinna, R. Broderson: The Biggascale Emulation Engine; FPGA 2002

/33/ A. Ast, et al.: Data-procedural Languages for FPL-based Machines; Proc. FPL'94

/34/ M. Herz et al.: (invited paper): Memory Organization for Data-Stream-based Reconfigurable Computing; Proc. ICECS 2002,

/35/ J. Becker: A Partitioning Compiler for Computers with Xputer-based Accelerators; Ph. D. dissertation, Kaiserslautern University, 1997

/36/ F. Catthoor et al.: Data Access and Storage Management for Embedded Programmable Processors; Kluwer, 2002

/37/ F. Catthoor et al.: Custom Memory Management Methodology Exploration of Memory Organization for Embedded Multimedia Systems Design; Kluwer, 1998

/38/ P. Kjeldsberg, F. Catthoor, E. Aas: Data Dependency Size Estimation for use in Memory Organization; IEEE Trans. on CAD, 22/5 (July 2003)

/39/ M. Weber et al.: MOM - Map Oriented Machine; in: E. Chiricozzi, A. D'Amico: Parallel Processing and Applications, North-Holland, 1988

/40/ H. Reinig et al.: Novel Sequencer Hardware for High-Speed Signal Processing; Proc. Design Methodologies for Microelectronics, Smolenice, Slovakia, Sept.1995

/41/ A. Hirschbiel et al.: A Flexible Architecture for Image Processing; Microprocessing and Microprogramming, vol 21, pp 65-72, 1987

/42/ M. Weber et al.: MoM - a partly custom-designed architecture compared to standard hardware; IEEE CompEuro 1989

/43/ M. S. Lam: Software Pipelining: an effective scheduling technique for VLIW machines; ACM SIGPLAN Conf. PLDI, 1988

/44/ L. Lamport: The Parallel Execution of Do-Loops; CACM 17,2, Feb. 1974

/45/ D. Loveman: Program Improvement by Source-to-Source Transformation; J.ACM Jan 1977

/46/ W. Abu-Sufah et al.: On the Performance Enhancement of Paging Systems Through Program Analysis and Transformations; IEEE-Trans. C-30(5), (May 1981)

/47/ J. Allen, K. Kennedy: Automatic Loop Interchange; Proc. ACM SIGPLAN'84 Symp. on Compiler Construction, June 1984

/48/ K. Schmidt et al.: Automatic Parallelism Exploitation for FPL-based Accelerators; HICSS'98

/49/ N. Petkov: Systolic Parallel Processing; North-Holland; 1992

/50/ M. Budiu and S. Goldstein: Fast Compilation for Pipelined Reconfigurable Fabrics; FPGA'99

/51/ I. Page, W. Luk: Compiling occam into FPGAs; Proc. FPL 1991

/52/ J. Becker et al.: A General Approach in System Design Integrating Reconfigurable Accelerators; Proc. IEEE ISIS'96; Austin, TX, Oct. 9-11, 1996

/53/ M. Herz, et al.: A Novel Sequencer Hardware for Application Specific Computing; . ASAP'97

/54/ M. Herz: High Performance Memory Communication Architectures for Coarsegrained Reconfigurable Computing Systems; Dissertation 2001, Univ. Kaiserslautern

/55/ R. Hartenstein et. al. (invited reprint): A Novel ASIC Design Approach Based on a New Machine Paradigm; IEEE J.SSC, Volume 26, No. 7, July 1991

/56/ R. Hartenstein (keynote address): Data-Stream-based Computing and Morphware; Joint 33rd Speedup and 19th PARS workshop; Basel, Switzerland, March 2003

/57/ N. Ebisuzaki et al.; 1997 Astrophysical Journal, 480, pp. 432,

/58/ R. Männer, R. Spurzem et al.: AHA-GRAPE: Adaptive Hydrodynamic Architecture - GRAvity PipE; Proc. FPL 1999

/59/ J.Becker (invited paper): Configurable Systems on Chip; Proc. ICECS 2002

/60/ J. Rabaey (keynote): Silicon platforms for the next generation wireless systems; FPL 2000

Reiner Hartenstein
Kaiserslautern University of 'Technology, Germany
http://hartenstein.de

Prispelo (Arrived): 15.09.2003 *Sprejeto (Accepted): 03.10.2003*

CONFIGURABILITY FOR SYSTEMS ON SILICON: REQUIREMENT AND PERSPECTIVE FOR FUTURE VLSI SOLUTIONS

Jürgen Becker

Universität Karlsruhe (TH), Institut für Technik der Informationsverarbeitung (ITIV), Karlsruhe, Germany

INVITED PAPER
MIDEM 2003 CONFERENCE
01.10.2003 - 03.10.2003, Grad Ptuj

Abstract: Systems-on-Chip (SoC) has become reality now, driven by fast development of CMOS VLSI technologies. Complex system integration onto one single die introduce a set of various challenges and perspectives for industrial and academic institutions. Important issues to be addressed here are cost-effective technologies, efficient and application-tailored hardware/software architectures, as well as corresponding IP-based EDA methods. This contribution will provide an overview on recent academic and commercial developments in Configurable Systems-on-Chip (CSoC) architectures, technologies and perspectives in different application fields, e.g. mobile communication and multimedia systems. Due to exponential increasing CMOS mask costs, essential aspects for the industry are adaptivity of SoCs, which can be realized by integrating reconfigurable re-usable hardware parts on different granularities into Configurable Systems-on-Chip (CSoCs).

Konfiguracijski sistemi na siliciju : Zahteve in vidiki za bodoča VLSI vezja

Izvleček: Zaradi hitrega razvoja CMOS VLSI tehnologij so dandanes sistemi na čipu (SoC) že realnost. Zapletene sistemske integracije na eno samo silicijevo tabletko predstavljajo vrsto izzivov za industrijske in akademske ustanove. V mislih imamo poceni tehnologije, učinkovite in uporabniško naravnane programske in strojne rešitve, kakor tudi odgovarjajoče metode elektronskega načrtovanja na osnovi intelektualne lastnine. Prispevek podaja pregled nad akademskim in komercialnim razvojem arhitektur konfiguracijskih sistemov na čipu (SoC) ter pregled nad pričakovanji in razvojem tehnologij na različnih področjih uporabe kot so mobilna telefonija in multimedijski sistemi. Zaradi visokih cen mask za CMOS tehnologije je prilagodljivost SoC sistemov bistvenega pomena za uporabo v industriji. To dosežemo z integracijo rekonfiguracijskih celic različne granulacije v konfiguracijski sistem na čipu (CSoC).

1. Introduction

Due to today's CMOS integration dimensions several designs and implementations of complex systems on silicon, so-called Systems-on-Chip (SoC), have been realized successfully. The term SoC is still not clearly defined and used with various interpretations in different situations. From my point of view, a SoC consists of at least two or more microelectronic macro-components of complexities previously integrated separately into different single dies. Thus, such components, also often called IP-cores (Intellectual Property), can be distinguished by one or more of the following criteria, characterizing also the major aspects of SoC-level integration decisions (see figure 1):

- integration technology, e.g. different MOS-/Bipolar transistors and materials (Si, SiGe, GaAs, etc.), electronic/mechanical systems (MEMS), etc.
- signal domain, e.g. digital, analog design style, e.g. full-custom, semi-custom, pre-diffused, pre-wired + non-MOS styles
- computing domain, e.g. processor (time domain), dedicated ASIC-based (space domain), dynamically reconfigurable (time / space domain) + various memory-cores and technologies
- specification and programming method, e.g. high-level language HLL (C, C++, SystemC, Matlab, Java, etc.), Assembler language (µC-specific), hardware description language HDL (Verilog /38/, VHDL /37/, Ella /41/, KARL /39/ /40/).

Thus, SoC-technologies are the consequent continuation of the ASIC technology, whereas complex functionalities, that previously required heterogeneous components to be merged onto a printed circuit board, are integrated within one single silicon chip. The first SoCs appeared in the early 1990s and consisted almost exclusively of digital logic constructions. Today SoCs are often mixed-technology designs, including such diverse combinations as embedded DRAM, high-performance or low-power logic, analog, RF, and even more unusual technologies like Micro-Electro-Mechanical Systems (MEMS) and optical input/output. But this development also raises its problems, e. g. it takes an enormous amount of time and effort (-> cost) to design and integrate a chip. The cornerstone of the required change in design methodologies will be the augmented use of parts from previous designs and by making use of parts designed by third parties, which is called IP- or Core-based design /10/ /15/ /16/. Dependent on application constraints, important aspects for SoC solutions are:

- time-to-market constraints have to be fulfilled,
- SoC architecture flexibility, e.g. risk minimization by adaptivity for application implementation, e.g. in cases of late specification changes,
- long product life cycles, due to multi-standard/multi-product implementation perspectives, and multi-purpose usage to fabricate high volumes of the same SoC (-> cost decrease per chip).

Recently, in addition to ASIC-based, one new promising type of SoC architecture template is recognized by several academic /4/ /31/ /32/ /28/ /29/ /30/ and first commercial versions /17/ /18/ /19/ /21/ /23/ /24/ /25/: Configurable SoCs (CSoCs), consisting of processor-, memory-, probably ASIC-cores, and on-chip reconfigurable hardware parts for customization to applications. CSoCs combine the advantages of both: ASIC-based SoCs and multichip-board development using standard components, e.g. they require only minimal NRE costs, because they don't need expensive ASIC-tools for developing always different and in the future very expensive mask sets, every time the functionality or standards are changing. Thus, besides other advantages, an enormous cost and risk minimization perspective is obvious for industrial CSoCs.

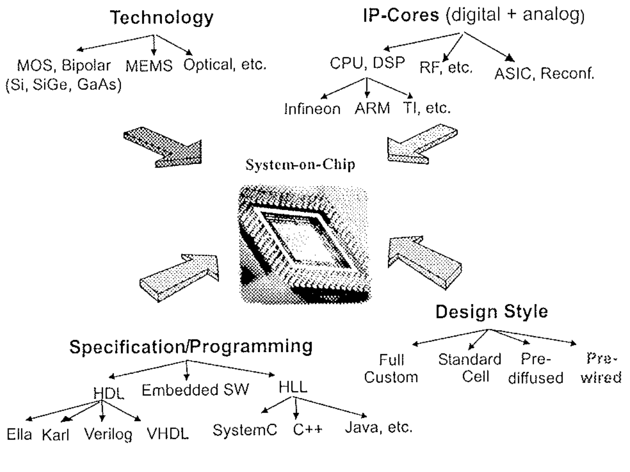


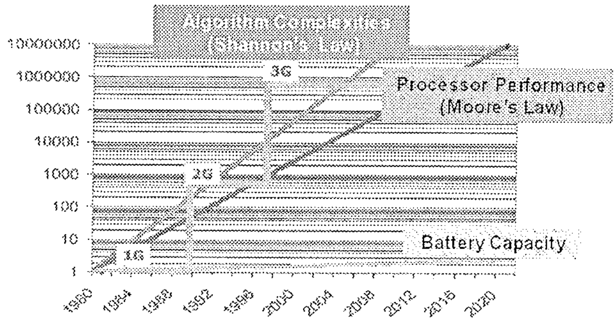
Fig. 1: SoC-level Integration Design Spacea

In the following, recent fine- and coarse-grain reconfigurable technologies as well as corresponding academic and commercial developments in architectures and applications are discussed. Reconfigurable hardware architectures have been proven in different application areas /11/ /12/ /32/ /17/ /18/ to produce at least one order of magnitude in power reduction and increase in performance. The focus of this contribution will describe the actual status and results of an industrial/academic CSoC integration, consisting of a SPARC-compatible LEON processor-core, a promising commercial coarse-grain XPP-array of suitable size from PACT XPP Technologies AG (Muenchen, Germany), and application-tailored global/local memory topology with efficient multi-layer Amba-based communication interfaces. The XPP architecture is regular structured for arbitrarily sized implementations, including regularity in combination with locality of data processing, e.g. for reducing power consumption. The complete adaptive SoC architecture is synthesized onto 0.18 and 0.13 µm UMC CMOS technologies at University of Karlsruhe (TH). Due to exponential increasing CMOS mask costs, the essential aspects for the industry are now risk-minimizing adaptivity and low cost of SoCs, which can be realized by integrating reconfigurable re-usable hardware parts on different granularities into CSoCs. In the last years ASIC/SoC markets for computer and communication applications had explosive revenue increases, compared to industrial and automotive areas. Relative to GSM, UMTS and IS-95 will require intensive layer 1 operations, which cannot be performed on today's processors /26/ /27/. Thus, optimized Hw/Sw partitioning of such computation-intensive tasks is necessary, whereas the flexibility to adapt to changing standards and different operation modes has to be considered. Based thereupon and future market demands, now several industrial and academic CSoC approaches arise /17/ /18/ / 19/ /21/ /22/ /23/ /25/ /28/ /29/ /30/ /31/ /32/.

2. Reconfigurable Technologies and Power/Cost Trade-offs

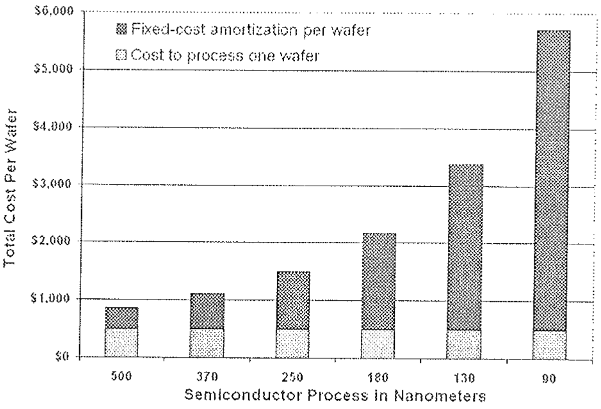
Today's processing requirements are rapidly increasing as well as changing for embedded electronic systems, e.g. in emerging applications like mobile communications, multimedia, automotive infotainment, telemetry and others, performance demands are growing rapidly. With the growth rate recently slowing down, the integration density of microprocessors is more and more falling back behind Moore's law. Accelerators occupy most of the silicon chip area. Compared to hardwired accelerators more flexibility is provided by (dynamically) reconfigurable hardware parts, which will be explained later.

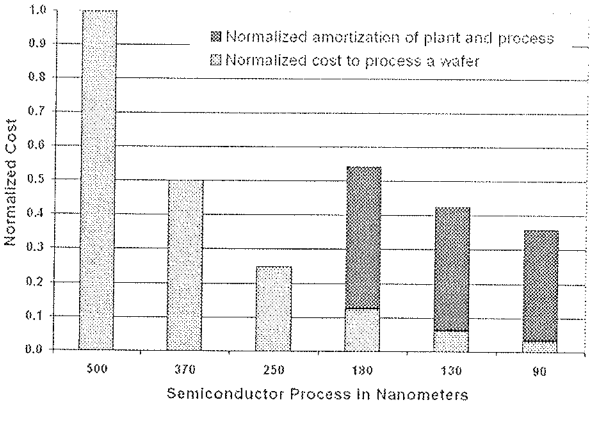
The low power optimization requirements are becoming more and more critical, either in the processor and especially in the embedded system world. The capacity of batteries is growing extremely slow (doubling every 30 years), especially compared to the increasing algorithm complexity and performance requirements, e.g. in future wireless algorithms (see figure 2). On the other side, the estimated processor performance and power figures cannot fulfill these requirements as well as the memory throughput demands, e.g. only every 10 years the growth of memory communication bandwidth is doubled. Because of the von Neumann bottleneck, memory bandwidth is an important issue. Avoiding this memory bottleneck not only by using accelerators, but also by innovative computing architectures, or even by breaking the dominance of the von Neu-



Signal Processing Algorithm (384 kbs)	DSP-Load [MIPs]
Digital Filter (RRC, Channelization)	~ 3600
Searcher (frame, slot, delay path est.)	~ 1500
RAKE Receiver	~ 650
Maximal Ratio Combining (MRC)	~ 24
Channel Estimation	~ 12
Turbo-Coding	~ 52
Total	**~ 5838**

Fig. 2: Future Wireless Applications: Algorithm Complexity vs. Performance vs. Power Trade-offs



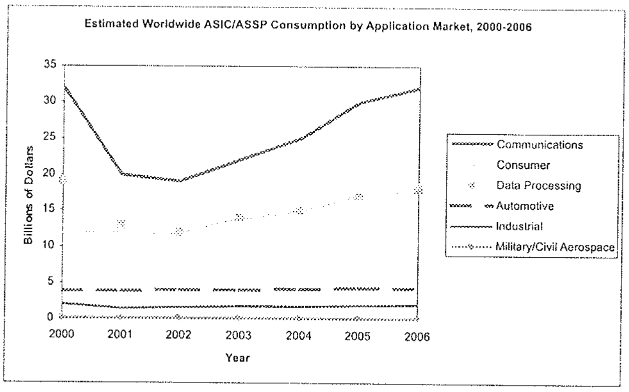


Source: Gilder Technology Report (Nick Tredennick , USA, 2003)

Fig. 3: Rising Costs per Wafer and the Amortization for Buildings and Equipment /33/

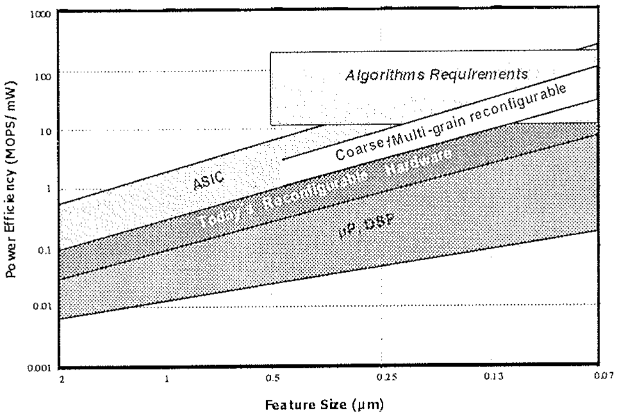
mann machine paradigm is a promising goal of new trends in embedded system development and CSE education.

Another, maybe most important aspect, is the exponential increase of CMOS mask costs, which results in an essential risk and cost factors for all development and production lines, e.g. smaller does in the future not necessarily mean better and cheaper. Moore´s law meant doubling the number of transistors per die every eighteen months, which results in more transistors on the same silicon area at equivalent costs, or in the same number of transistors at lower costs. This theory assumes the downscaling of transistor dimensions by √2 every eighteen months and that the cost to process a wafer depends mainly on its size. This "law" was fulfilled by the corresponding semiconductor industry for a long time this way and returned the expected efficiency. Unfortunately, we have to deal now with a different situation, because the fixed costs for a semiconductor plant and to process a wafer have been increased exponentially in the last years, e.g. the lithography equipment and the cost per wafer mask set. This results in very high fixed cost factors for each wafer compared to the relative small variable costs to process a wafer through a fab line. The corresponding process and cost interrelations were evaluated and quantized by Nick Tredennick in his Gilder Technology Report /33/. In figure 3 a) the exponentially rising wafer fixed costs and the variable wafer processing costs are illustrated dependent on the transistor technologies, and figure 3 b) shows the cheapest transistors to be fabricated by fully amortized 250 nm fabrication lines. The assumptions in figure 3 do not consider the tremendous and even more increasing mask set costs, so that smaller transistors will be even more expensive. For more details about actual changes in semiconductors, especially about the detailed quantization formulas and assumptions and finally resulting process adoption rates, please see /33/. The former dominance of the procedural von Neumann microprocessor paradigm has been due to its RAM-based flexibility and that in many cases no application-specific silicon is needed.



Source:
Gartner Dataquest 2002

Fig. 4: ASIC/ASSP Semiconductor Consumption of different Application Areas



Source: T. Claassen (ISSCC '99)
and R. Hatenstein (ICECS 2002)

Fig. 5: Energy / Flexibility Conflict of different Hardware Architectures and Circuits

Throughput is the only limitation because of its sequential nature of operation. But now a second RAM-based computing paradigm is heading for mainstream: the application of multi-grain (dynamically) reconfigurable hardware architectures. Such kind of structural programming in space - in contrast to von Neumann based programming in time - provides massive parallelism at logic, operator and arithmetic level, often more efficient than vN-based process level parallelism. As a consequence of all facts and views described above we have to target new ways in exploiting the available silicon and technologies, e.g. not always the newest and most highly integrated versions, in more effective way. To fulfill the cost, power as well as performance requirements of today's and future algorithm complexities new computing architectures and circuits with more efficiency, flexibility and operation cleverness have to be developed and applied. Thus, today's fine-grain and especially coarse- as well as multi-grain (dynamically) reconfigurable architectures will realize better performance / energy trade-offs than comparable mp, DSP or µController platforms (see figure 5). Moreover, their (online) flexibility and silicon re-use features will result in essential cost and risk minimization effects necessary for future processor, VLSI and System-on-Chip solutions. The application fields and with corresponding complex algorithms and estimated ASIC/ASSP consumption are illustrated in figure 4.

The following section gives an overview on some selected industrial and academic architectures and System-on-Chip solutions applying fine- and coarse-grain (dynamically) reconfigurable hardware datapaths for several of the above mentioned algorithm fields.

3. Academic and Industrial System-on-Chip Solutions

Today's fine-grain and early coarse-grain reconfigurable hardware architectures are very useful in several application fields and are alternatives to specialized (multi-) processor solutions /4/ /7/ /8/ /10/ /11/ /12/ /13/ /17/ / 18/ /28/ /29/ /30/ /32/ /34/. But, a minor part of the fine-grain area is used by CLBs (configurable logic blocks), which are the logic resources. Major part of the area is covered by a reconfigurable interconnect fabrics, provid-

Source: R. Hartenstein

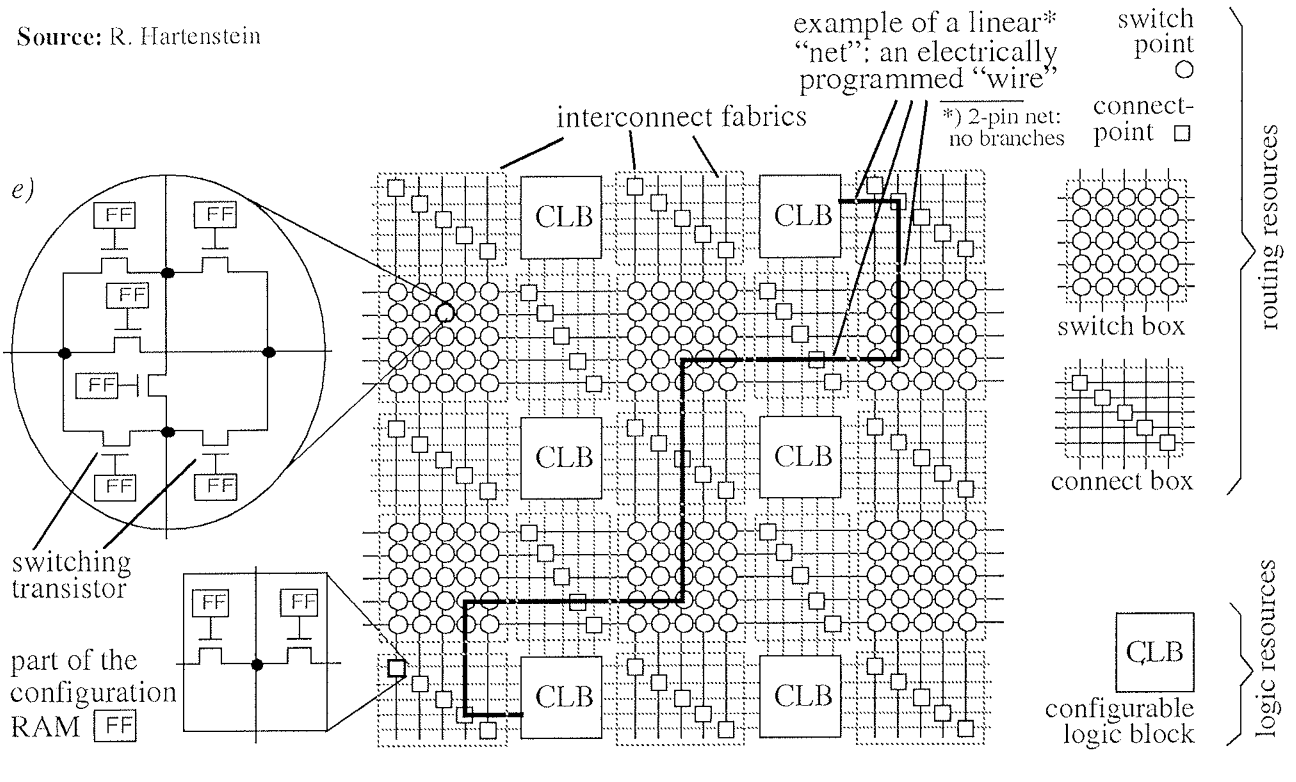
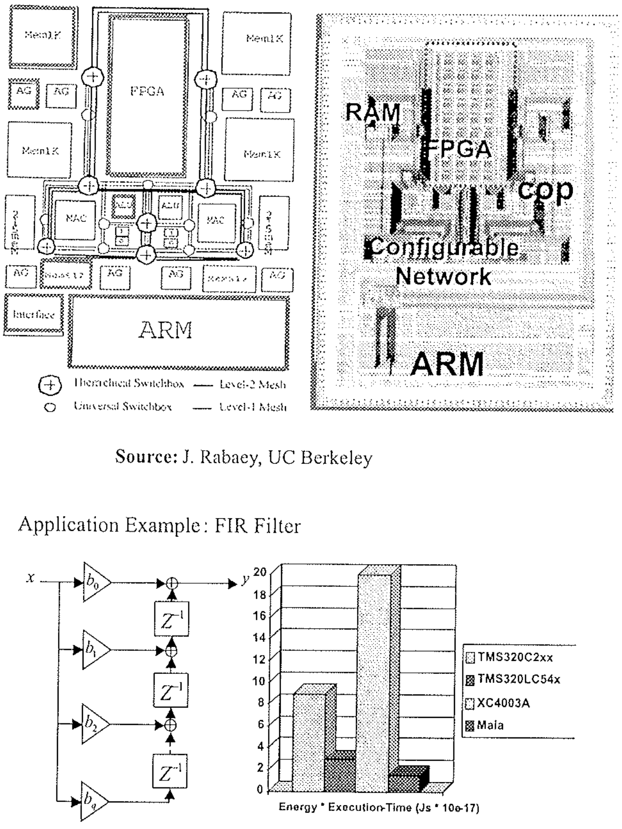


Fig. 6: Illustration of fine-grain reconfigurable hardware resources (FPGA: 1 configured "wire" shown) /34/

ing wire pieces, switch boxes, and connect boxes to connect a pin of a CLB, with a pin of another CLB by programming a "soft wire" (an example shown in figure 6). The state of each switching transistor is controlled by a Flip-flop (FF) which is part of "hidden" configuration RAM (not shown in figure 6), also used to program the CLBs to select the particular logic function of each. By downloading new configuration code all this can be re-programmed anywhere and at any time. In the following an efficient fine-grain System-on-Chip solution tailored to baseband voice coding algorithm will be sketched. Within the MAIA CSoC a fine-grain FPGA-core realizes the reconfigurable hardware part. In general, the MAIA architecture consists of one control processor and other satellite units (can be processors, FPGAs or other units such as MAC, see figure 7). During computation and reconfiguration sequential threads are instantiated on the control processor, which configures the satellite processors and the on-chip reconfigurable communication network and manages the overall control flow of applications, either in a static compiled order, or through a dynamic real-time kernel. Thus, the architecture is reconfigurable in two respects - inter-satellite communication configurations and the fine-grain FPGA hardware part. The MAIA processor consists of a microprocessor core (ARM8) and 21 satellite processors: two MACs, two ALUs, eight address generators, eight embedded memories (4 512x16bit, 4 1kx16bit) and an embedded low-energy FPGA. Connections between satellites are accomplished through 2-level hierarchical mesh-structured reconfigurable interconnect network. The ARM8 uses an interface control unit to configure and communicate data with satellites. The address generators and embedded memories are distributed to supply multiple parallel data streams to the computational elements. The MAIA chip was implemented using 0.25U 6-level metal CMOS process with a supply voltage of 1V and additional voltages of 0.4V and 1.5V, The die size of the implementation was 5.2mm x 6.7mm with 1.2 million transistors at 40 MHz with an average power dissipation of 1.5-2 mW. The Maia CSoC is optimized for selected mobile communication application parts, e. g. a full-rate VSELP voice coder algorithm was implemented at 30 MHz with 5.7 GOPS/Watt /31/.



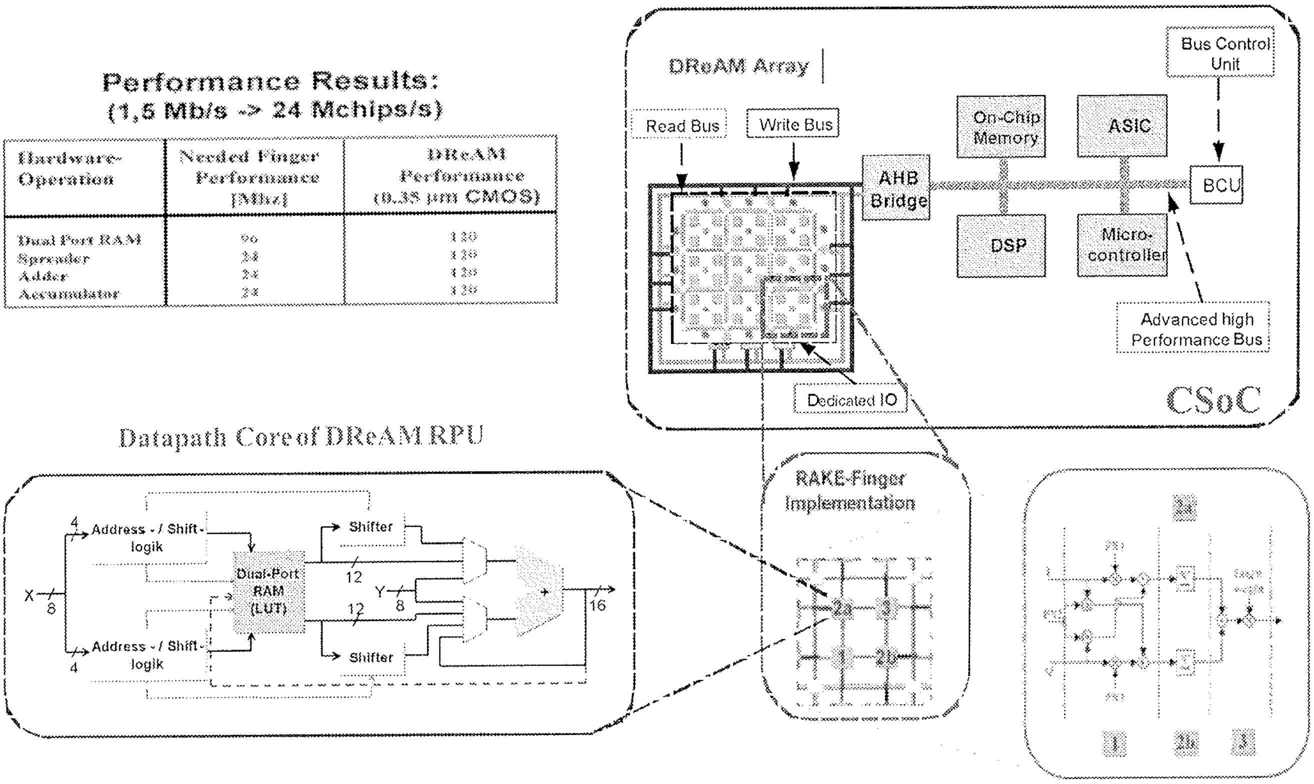
Source: J. Rabaey, UC Berkeley

Application Example: FIR Filter

Fig. 7: MAIA CSoC and FIR Application /31/ /32/

Fine grain morphware lacks area/power-efficiency (figure 6). The physical integration density (transistors per chip) of FPGAs is roughly 2 orders of magnitude worse than the Gordon Moore Curve. Due to reconfigurability overhead roughly about only one percent of these transistors deserve the real application, so that the logical integration density is about 4 orders of magnitude behind Gordon Moore. For high throughput requirements coarse-grain reconfigurable hardware is the much more powerful and more area-efficient, also providing a massive reduction of embedded memory and time needed for configuration /34/. Coarse grain morphware is also about one order of magnitude more energy-efficient than fine-grain solutions (figure 5 and /1/ /2/ /3/). Whereas fine-grain FPGAs are using single bit wide CLBs (figure 6), coarse-grain reconfigurable Computing uses RPUs (reconfigurable processing units), which, similar to ALUs, have major path widths, like 32 bits, for instance. Important applications stem from the performance limits of the "general purpose" processor, creating a demand for accelerators. Especially in application areas like multimedia, wireless telecommunication, data communication and others, the throughput requirements are growing faster than Moore's law (growth of required bandwidth: figure 2), along with growing flexibility requirements due to unstable standards and multi-standard operation /4/. Currently the requirements can be met only by coarse-grain hardware arrays from a provider like PACT (figure 9 and /5/).

First, a second selected academic CSoC example will be sketched here. This is an application-tailored architecture called DReAM /4/ /14/, a coarse-grain Dynamically Reconfigurable Architecture for Mobile communication systems. It was designed at the Darmstadt University of Technology for the requirements of future mobile communications systems. Especially the application area of mobile communication requires an adaptable SoC solution. The total system view of such a CSoC is shown in figure 8 /4/. The datapath oriented DReAM array can be seen in figure 8. It consists of an array of coarse-grained, dynamically Reconfigurable Processing Units (RPUs), which are connected with a local and a global communication network. The RPU is the major hardware component of the DReAM, which executes mainly arithmetic data manipulations for signal processing parts. In addition, dual-port



Performance Results:
(1,5 Mb/s -> 24 Mchips/s)

Hardware-Operation	Needed Finger Performance [Mhz]	DReAM Performance (0.35 µm CMOS)
Dual Port RAM	96	120
Spreader	24	120
Adder	24	120
Accumulator	24	120

Fig. 8: *DReAM CSoC Architecture Datapath and RAKE Application Results /4/*

RAMs are used as Look-Up Tables when performing multiplications and the application-specific units are used for PN-code correlation operations. The DReAM architecture provides efficient and fast dynamic reconfiguration possibilities, e.g. only partly and during runtime. Further details to implemented examples and mapping techniques as well as performance results, e.g. a RAKE-Receiver specification for a data rate of 1.5 Mb/s based on a 0.35 µm CMOS-process, can be found in /4/ /14/.

Next, two commercial CSoC solutions will be described:

- the A7 architecture from Triscend with fine-grain on-chip reconfigurable hardware /19/ /20/
- the dynamically reconfigurable XPP Architecture from PACT /23/, /24/, /6/, /7/

The A7 Configurable System-on-Chip (CSoC) device /19/, /20/ is a complete, high-performance user-programmable system, which contains an embedded 32-bit ARM7TDMI RISC processor and an embedded programmable logic architecture, optimized for processor and bus interface, a high-performance 32-bit internal bus supporting up to 455M-bytes per second peak transfer rates, and 16K-bytes of internal scratchpad SRAM memory and a separate 8K-byte cache. The ARM7TDMI is a general-purpose 32-bit RISC microprocessor that supports the complete ARM 32-bit instruction set and the reduced 16-bit instruction set. The ARM processor is integrated with other system components and the Configurable System Logic (CSL) matrix to provide a complete CSoC system. The embedded SRAM-based Configurable System Logic (CSL) matrix provides full, easy-to-use system customization. The high-performance programmable logic architecture consists of a highly interconnected matrix of CSL cells. Resources within the matrix provide seamless access to and from the internal high-performance Configurable System Interconnect (CSI) bus, interconnecting the embedded processor, its peripherals, and the CSL matrix at a maximum speed of 60MHz. Each CSL cell performs various potential functions, including combinatorial and sequential logic and the output blocks (PIOs) provide a highly flexible interface between external functions and the internal system bus.

A very interesting and promising approach for CSoC integration is the eXtreme Processing Platform (XPP) /23/ /24/, /6/ /7/ (see figure 9), realizing a new runtime reconfigurable data processing technology that replaces the concept of instruction sequencing by configuration sequencing with high performance application areas envisioned from embedded signal processing to co-processing in different DSP-like application environments. The adaptive reconfigurable data processing architecture consist of following components:

- Processing Array Elements (PAEs), organized as Processing Arrays (PAs),
- a packet oriented communication network,
- a hierarchical Configuration Manager (CM) tree, and
- a set of I/O modules.

This supports the execution of multiple data flow applications running in parallel. A PA together with one low level

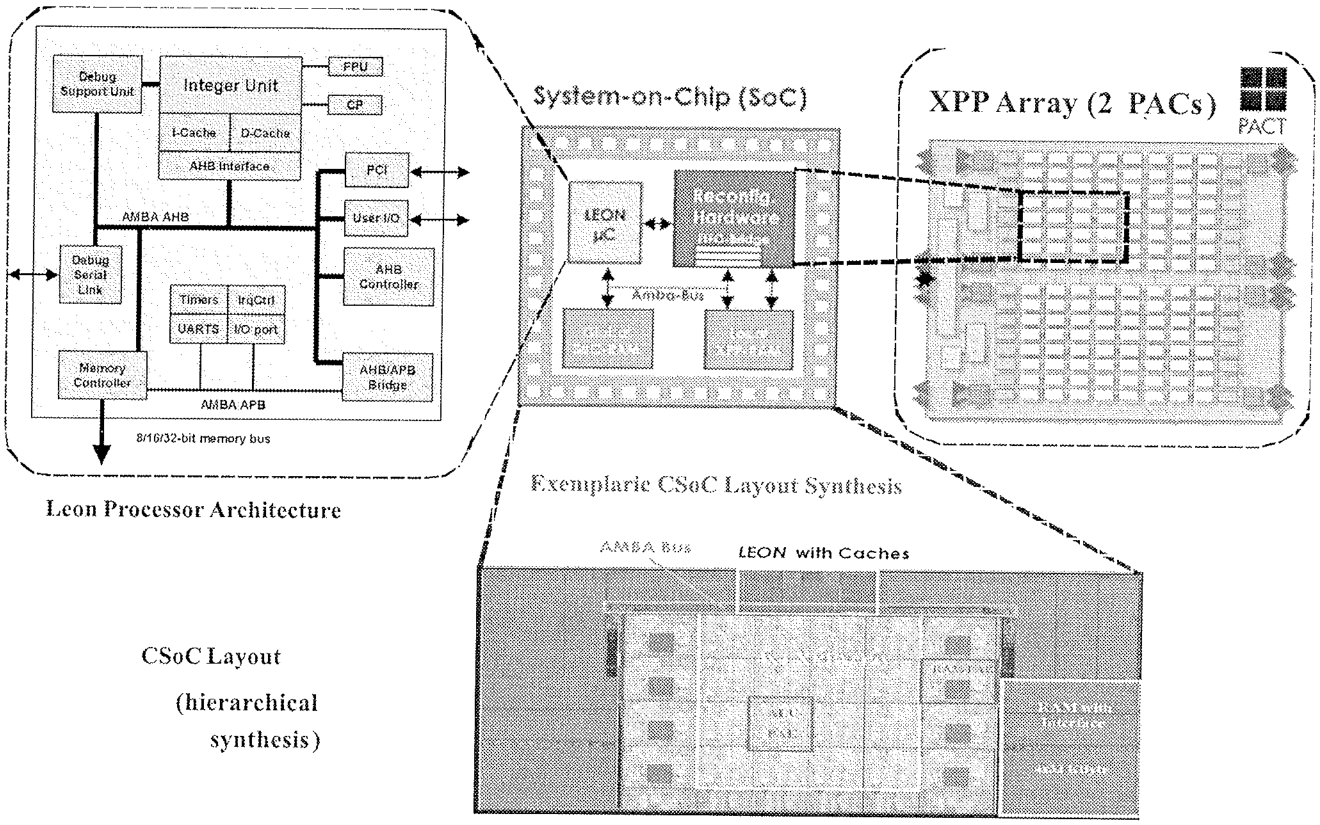


Fig. 9: PACT-/Leon-based CSoC Architecture and Layout synthesized at Universitaet Karlsruhe (TH)

CM is referred as PAC (Processing Array Cluster). The low level CM is responsible for writing configuration data into the configurable objects of the PA. Typically, more than one PAC is used to build a complete XPP device. Doing so, additional CMs are introduced for configuration data handling. With an increasing number of PACs on a device, the configuration hardware assumes the structure of a tree of CMs. The root CM of the tree is called the supervising CM or SCM. This unit is usually connected to an external or global RAM. The basic concept consists of replacing the Von-Neumann instruction stream by automatic configuration sequencing and by processing data streams instead of single machine words, similar to /12/. Due to the XPP´s high regularity, a high level compiler can extract instruction level parallelism and pipelining that is implicitly contained in algorithms /6/. The XPP can be used in several fields, e.g. as image/video processing, encryption, and baseband processing of next generation wireless standards, e.g. to realize also Software Radio approaches. 3G systems, i.e. based on the UMTS standard, will be defined to provide a transmission scheme which is highly flexible and adaptable to new services. Relative to GSM, UMTS and IS-95 will require intensive layer 1 related operations, which cannot be performed on today's processors /26/ /27/. Thus, an optimized HW/SW partitioning of these computation-intensive tasks is necessary, whereas the flexibility to adapt to changing standards and different operation modes (different services, QoS, BER, etc.) has to be considered. Therefore, selected computation-intensive signal processing tasks have to be migrated from software to hardware implementation, e.g. to ASIC or coarse-grain reconfigurable hardware parts, like the XPP architecture. Within the application area of future mobile phones desired and important functionalities are gaming, video compression for multimedia messaging, polyphone sound (MIDI), etc. Therefore, a flexible, low cost hardware platform with low power consumption is needed for realizing necessary computation-intensive algorithms parts. Thus, PACT implemented several of these functionalities onto the cost-efficient 4x4 XPP array size, e.g. a 256-point FFT, a real 16 tap FIR filter, and a video 2d DCT (8x8) for MPEG-4 systems. Their newest commercial CSoC is called SMeXPP and consists of a an ARM-7 EJS and an 4x4 XPP array with efficient RAM-topologies promising a high boost in performance and flexibility. The technical and commercial trade-offs of this SMeXPP solution is described in /7/ and /8/. First digital TV application performance results were obtained by evaluating corresponding MPEG-4 algorithm mappings onto the introduced ARM/XPP CSoC and based on the 0.13 µm CMOS technology synthesis results. Based on this coarse-grain CSoC version, performance/cost results of an MPEG-4 application is currently under implementation, whereas the Inverse DCT applied to 8x8 pixel blocks can be performed by an 4x4 XPP-Array in 74 clock cycles. Since the IDCT is one of the most complex operations in MPEG-4 algorithms, the preliminary clock frequency of 100 MHz based on 0.13 µm CMOS technology integration is sufficient for this real-time digital TV application scenario.

Another class of resources for reconfigurable computing is called multi-grain reconfigurable hardware, where several fine-grain pathwidth slices (2-/4-bits, for instance) with slice bundling capability including carry signal propagation can be configured to be merged into RPUs with a pathwidth of multiples of the slice path width (e. g. 16, 20, or 24 bits). Moreover, dependent on the targeted algorithm classes, bitlevel data operations, wordlevel arithmetic instructions, or even control-driven FSMs should be supported. These new hybrid architectures, combining the advantages of fine- and coarse-grain circuits into novel generic datapath approaches, are currently under development in different specialized research programs, e.g. funded by the German DFG and other institutions /35/.

4. SoC Education Aspects

The challenges in the development of application-tailored SoCs influences and changes the traditional design flow for chip, and thus today's engineering education. This should have some impact to the way how students in electronic engineering departments are taught, e.g. courses which fully cover all required skills for a SoC designer. The traditional education for students enables them to design stand-alone hardware components such as ASICs, instruction-set processor, memory, FPGA, analog and even RF CMOS chips /36/. Specially educated engineers are responsible for combining these components to a system. With the upcoming of SoCs these till now completely separate categories of design will merge to one design flow. A chip will no longer be assembled at the gate level but at the IP block level and IP interfaces /36/. Multidisciplinary system thinking is required for future designs, e.g. a vertical integration of system and application know-how with CAD and technology knowledge has to be realized in vertical education projects and labs (see figure 10). This education goal could be achieved successfully by the co-working of students and faculty within real system design projects, formalizing and encapsulating application-specific techniques into reusable methods, libraries and tools shared by the entire educational community. Students and universities need access to the latest technical and industrial developments, and education has to be focused also on techniques and theories which are fundamental and time invariant. Such system architects /36/ should be able to operate efficiently in interdisciplinary teams with highly soft skilled members, required urgently by today's embedded systems divisions.

5. Conclusions and Outlook

The paper has given an introduction and overview on reconfigurable hardware systems and their VLSI integration. It also has pointed out future trends driven by technology progress and EDA innovations. Many system-level integrated future products without reconfigurability will not be competitive. Instead of continuous technology progress and deep-submicron integration more efficient and clever architectures by (dynamically) reconfigurable platform usage will often be the key to keep up the current innovation speed beyond the technology limits of silicon. It is time to revisit the available scientific results from reconfigurable-related R&D to derive promising commercial solutions and corresponding curricular updates in EE and CS education. Exponentially increasing CMOS mask costs demand adaptive and re-usable silicon, which can be efficiently realized by integrating reconfigurable circuits of different granularities into CSoCs, providing a potential for short time-to-market and post-fabrication error/functionality corrections (risk minimization!), multi-purpose/-standard features including comfortable application updates within product life cycles (volume increase: cost decrease). This results in the fact that several major industry players are currently integrating (dynamically) reconfigurable cores/datapaths into their processor architectures and system-on-chip solutions.

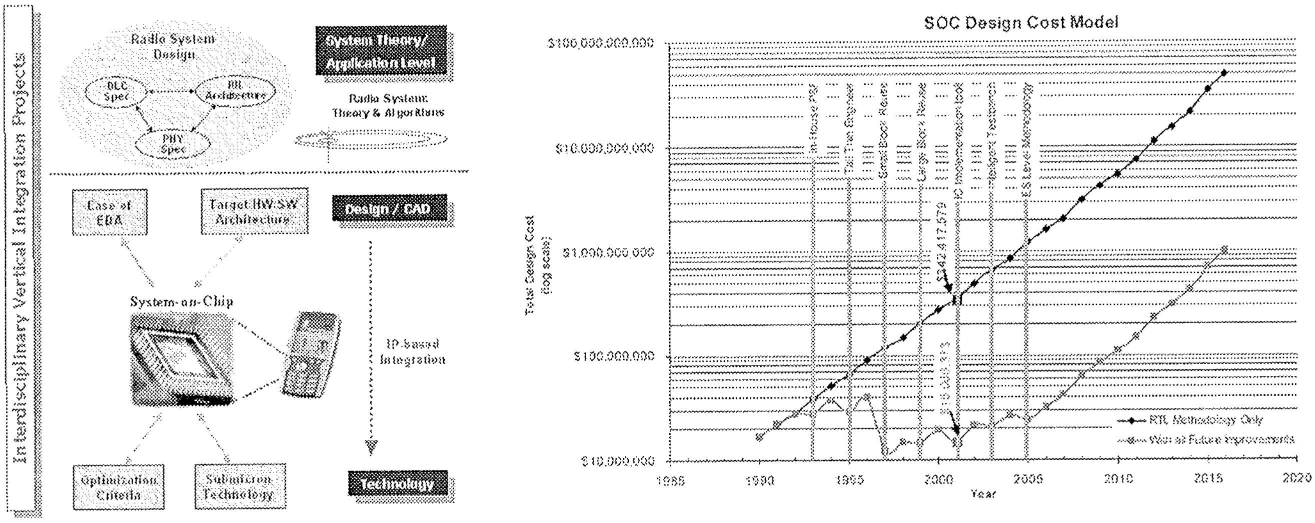


Fig. 10: CAD/VLSI Education Challenges and SoC Cost Aspects

6. References

/1/ R. Hartenstein (invited): The Microprocessor is no more General Purpose; Proc. ISIS 1997

/2/ R. Hartenstein (invited): Trends in Reconfigurable Logic and Reconfigurable Computing; ICECS 2002

/3/ A. DeHon: The Density Advantage of Configurable Computing; IEEE Computer, April 2000

/4/ J. Becker, T. Pionteck, M. Glesner: An Application-tailored Dynamically Reconfigurable Hardware Architecture for Digital Baseband Processing; SBCCI 2000

/5/ http://pactcorp.com

/6/ V. Baumgarte, et al.: PACT XPP - A Self-Reconfigurable Data Processing Architecture; ERSA 2001

/7/ J. Becker, M. Vorbach: Architecture, Memory and Interface Technology Integration of an Industrial/Academic Configurable System-on-Chip (CSoC); IEEE Computer Society Annual Workshop on VLSI (WVLSI 2003), Tampa, Florida, USA, February, 2003

/8/ M. Vorbach, J. Becker: Reconfigurable Processor Architectures for Mobile Phones; Reconfigurable Architectures Workshop (RAW 2003), Nice, France, April, 2003

/9/ "ASIC Sstem-on-a-Chip", Integrated Circuit Engineering (ICE), http:www.ice-corp.com

/10/ M. Glesner, J. Becker, T. Pionteck: Future Research, Application and Education Perspectives of Complex Systems-on-Chip (SoC); Proc. of Baltic Electronic Conference (BEC 2000), Oct. 2000, Tallinn, Estonia

/11/ P. Athanas, A. Abbot: Real-Time Image Processing on a Custom Computing Platform, IEEE Computer, vol. 28, no. 2, Feb. 1995.

/12/ R. W. Hartenstein, J. Becker et al.: A Novel Machine Paradigm to Accelerate Scientific Computing; Special issue on Scientific Computing of Computer Science and Informatics Journal, Computer Society of India, 1996.

/13/ J. Rabaey, "Reconfigurable Processing: The Solution to Low-Power Programmable DSP", Proceedings ICASSP 1997, Munich, April 1997.

/14/ J. Becker, N. Liebau, T. Pionteck, M. Glesner: Efficient Mapping of pre-synthesized IP-Cores onto Dynamically Reconfugurable Array Architectures; Proc. 11th Int´l Conference on Field Programmable Logic and Applications, Belfast, Ireland, 2001.

/15/ Y. Zorian, R. K. Gupta,: Design and Test of Core-Based Systems on Chips, it IEEE Design & Test of Computers, pp. 14-25, Oct.-Dec. 1997.

/16/ B. Tuck, Integrating IP blocks to create a system-on-a-chip, it Computer Design, pp. 49-62, Nov. 1997.

/17/ Xilinx Corp.: http://www.xilinx.com/products/virtex.htm.

/18/ Altera Corp.: http://www.altera.com

/19/ Triscend Inc.: http://www.triscend.com

/20/ Triscend A7 Configurable System-on-Chip Platform - Data Sheet http://www.triscend.com/products/dsa7csoc_summary.pdf

/21/ LucentWeb/ http://www.lucent.com/micro/fpga/

/22/ Atmel Corp.: http://www.atmel.com

/23/ PACT Corporation: http://www.pactcorp.com

/24/ The XPP Communication System, PACT Corporation, Technical Report 15, 2000

/25/ Hitachi Semic.: http://semiconductor.hitachi.com/news/triscend.html

/26/ Peter Jung, Joerg Plechinger., "M-GOLD: a multimode basband platform for future mobile terminals",CTMC'99, IEEE International Conference on Communications, Vancouver, June 1999.

/27/ Jan M. Rabaey: System Design at Universities: Experiences and Challenges; IEEE Computer Society International Conference on Microelectronic Systems Education (MSE´99), July 19-21, Arlington VA, USA

/28/ S. Copen Goldstein, H. Schmit, M. Moe, M. Budiu, S. Cadambi, R. R. Taylor, R. Laufer "PipeRench: a Coprocessor for Streaming Multimedia Acceleration" in ISCA 1999. http://www.ece.cmu.edu/research/piperench/

/29/ MIT Reinventing Computing: http://www.ai.mit.edu/projects/transit dpga_prototype_documents.html

/30/ N. Bagherzadeh, F. J. Kurdahi, H. Singh, G. Lu, M. Lee: "Design and Implementation of the MorphoSys Reconfigurable Computing Processor "; J. of VLSI and Signal Processing-Systems for Signal, Image and Video Technology, 3/ 2000

/31/ Hui Zhang, Vandana Prabhu, Varghese George, Marlene Wan, Martin Benes, Arthur Abnous, "A 1V Heterogeneous Reconfigurable Processor IC for Baseband Wireless Applications", Proc. of ISSCC2000.

/32/ Pleiades Group: http://bwrc.eecs.berkeley.edu/Research/Configurable_Architectures/

/33/ Nick Tredennick: Gilder Technology Report, vol. IX no. 4, April 2003, USA

/34/ J. Becker, R. Hartenstein: Configware and Morphware going Mainstream; Journal of Systems Architecture JSA (Special Issue on Reconfigurable Systems), June 2003

/35/ Deutsche Forschungsgemeinschaft (DFG): Specialized Research Program 1148 „Reconfigurable Computing Systems": http://www12.informatik.uni-erlangen.de/spprr/

/36/ H. De Man: System-on-Chip Design: Impact on Education and Research, IEEE Design & Test of Computers, publ. July-Sept. 1999, Volume 16 3, Page(s) 11-19

/37/ IEEE Standard VHDL Language Reference Manual, Institute of Electrical and Electronics Engineers Inc., 1994, ISBN 1-55937-376-

/38/ http://www.verilog.com

/39/ Hauck, R.: KARL-4 - A hardware description language for the design and synthesis of digital hardware; Proc. 2nd ABAKUS workshop, Innsbruck, Austria, Sept. 1988

/40/ R. Hartenstein: Hardware Description Languages; Elsevier, Amsterdam, 1987.

/41/ J D Morison, A S Clarke: ELLA 2000, A Language for Electronic System Design; McGraw-Hill Book Company, ISBN 0-07-707821-7

Jürgen Becker
Universität Karlsruhe (TH)
Institut für Technik der Informationsverarbeitung (ITIV)
D-76128 Karlsruhe, Germany
http://www.itiv.uni-karlsruhe.de/
becker@itiv.uni-karlsruhe.de

Prispelo (Arrived): 15.09.2003 *Sprejeto (Accepted): 03.10.2003*

UDK621.3:(53+54+621+66), ISSN0352-9045

DISTRIBUTED EMBEDDED SAFETY CRITICAL REAL-TIME SYSTEMS, DESIGN AND VERIFICATION ASPECTS ON THE EXAMPLE OF THE TIME TRIGGERED ARCHITECTURE

Manfred Ley; Christian Madritsch

Fachhochschule Technikum Kärnten, Carinthia Tech Institute, Villach, Austria

INVITED PAPER
MIDEM 2003 CONFERENCE
01. 10. 03 - 03. 10. 03, Grad Ptuj

Abstract: The Time Triggered Architecture (TTA) and its related communication protocol, TTP/C is an emerging communication principle for distributed fault-tolerant real-time systems. Typical applications are safety-critical digital control systems such as drive-by-wire and fly-by-wire.

This paper highlights the hardware / software architecture and design of the first industrial single chip communication controller for the Time Triggered Protocol (TTP/C). An application specific RISC core with several specialized peripheral blocks, RAMs, flash memory and analog cells was implemented together with necessary protocol firmware to fulfill both cost and safety requirements. Whereas the controller chip itself can be seen as an embedded system, the composability characteristic of TTA enables a hierarchical system design style with nodes and communication clusters as higher level system components embedded into an application device like a car or airplane. A complete framework for hardware / software co-simulation and verification across all levels of hierarchy was buildt up to support the design work from chip to system level. Furthermore, system reliability and fault behavior of a safety critical system has to be shown to safety certification authorities. Extensive fault injection experiments have been performed at simulation and physical level to proof the concept, fault model and resulting implementation of an embedded TTA control system.

Distribuirani vgrajeni varnostni sistemi v realnem času – zasnova in verifikacija na primeru časovno prožene arhitekture

Izvleček: Časovno prožena arhitektura (TTA) in njen odgovarjajoči komunikacijski protokol, TTP/C, je porajajoči se komunikacijski princip za distribuirane sisteme odporne na napake v realnem času. Tipična uporaba so digitalni sistemi za nadzor kot so vožnja in letenje krmiljeno s povezavo.

V prispevku prikažemo programsko in strojno arhitekturo ter načrtovanje prvega industrijskega komunikacijskega nadzornega veza za TTP/C protokol na enem čipu. Uporabniško specifično RISC jedro z večimi specializiranimi perifernimi bloki, RAMi, FLASH pomnilniki in analognimi celicami je bilo uporabljeno skupaj s potrebnimi komponentami TTC/P protokola z namenom zadovoljiti varnostnim in stroškovnim zahtevam. Čeprav na celoten nadzorni čip lahko gledamo kot na vgrajen sistem, pa sestavne karakteristike protokola TTA omogočajo bolj hierarhični stil načrtovanja sistema z vozlišči in komunikacijskimi skupki kot sistemskimi komponentami na višji ravni vgrajenimi v neko uporabniško okolje, kot sta npr. avtomobil ali letalo. Zgradili smo celotno okolje potrebno za simulacijo in verifikacijo programske in strojne opreme na vseh hierarhičnih nivojih z namenom podpreti proces načrtovanja od čipa do sistema. Oblastem pristojnim za potrjevanje ustreznosti varnosti sistema je bilo dodatno potrebno prikazati zanesljivost in vedenje sistema v primeru napak. Opravili smo obsežne poskuse s povzročanjem napak na fizičnem in simulacijskem nivoju z namenom dokazati pravilnost koncepta, modela napak in delovanja dokončne implementacije vgrajenega TTA nadzornega sistema.

1. Introduction

The Time-Triggered Architecture (TTA) is designed for a wide range of fault-tolerant distributed real-time systems /1/. The application domain of the architecture is safety-critical by-wire systems in the automotive, aerospace and railway industries.

The key component of the Time-Triggered Architecture is a VLSI communication controller, which executes the Time-Triggered Protocol (TTP) /2//3/ and provides all communication and safety features of TTP to a host controller running the application of a network node. Based on a prototype implementation from the Technical University of Vienna /4//5/, Carinthia Tech Institute (CTI) designed an industrial (automotive specification) single chip version of such a communication controller, the TTA-C2. Together with the Technical University of Vienna and TTTech AG a complete HW-SW codesign environment was developed. Furthermore, related projects like FIT (see chapter 7.2) were carried out to proof concept as well as implementation. This paper should give an overview on this development with a focus on safety related issues and parts of the design.

After an introduction of the TTA system principles in Section 2 we describe the controller architecture and its building blocks in Section 3. Section 4 explains the design flow applied and gives technical details of the new chip. Section 5 details the modeling strategy and tools for hardware/software co-development. The system development process and the proof of safety relevant system behavior are explained in Sections 6 and 7. Finally we summarize the work done within the TTA activities at CTI.

2. Time Triggered Architecture

In contrast to a usual event triggered system, where messages are initiated by events independent from the communication system, the Time-Triggered Architecture uses a predefined global TDMA schedule for all communication activities. Each message on the bus is only initiated by the progression of time according to the preplanned message timetable. In a distributed system all TTA nodes are synchronizing themselves to a common global time using the fault-tolerant average algorithm and can therefore communicate without conflicts. TTA nodes connected to a bus using the same global time are called a TTA cluster. An autonomous communication controller decouples the host (application) subsystem from the communication subsystem in both the logical and temporal domain. No control lines or interrupts connect the application processor to the TTP/C controller, the only interface is a dual ported RAM called Communication Network Interface (CNI). On the bus side a bus guardian circuit monitors access to the physical layer. This effectively prevents any fault propagation from a faulty node to the whole system. A basic TTA node consists of the communication controller TTA-C2 and a host CPU system (e.g. Infineon C167CR) running the nodes application. A block diagram for three nodes of a TTP/C cluster is shown in Figure 1. Figure 2 shows the time division bus access of each node.

The communication controller delivers fault-tolerant services according to the TTP/C protocol specification to the host subsystem. Most important features are the synchronized global clock, message transmission, network consistency check and membership service, specification details can be found in /6/. The redundant physical communication layer of TTP/C can be realized as two copper twisted pairs or fiber optics connection.

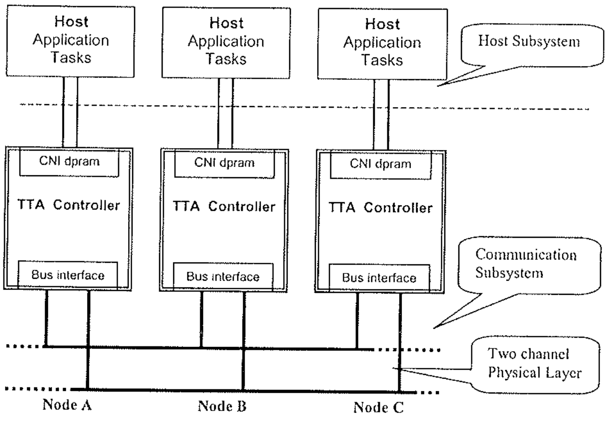


Figure 1: TTA Cluster Architecture

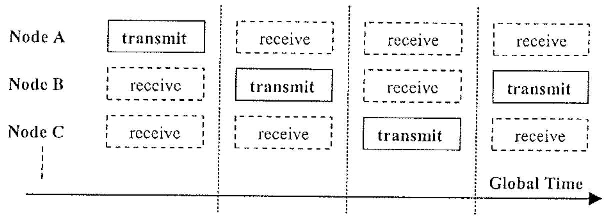


Figure 2: TTA Bus Access Schema

3. Communication Controller Hardware

3.1 Requirements

Discrete component implementations of the TTP/C protocol, using a micro controller together with FPGAs, lack in transmission bandwidth for industrial applications. Performance analysis shows that clock synchronization, data transmission, TTP bus timing, CRC calculation, and the bus guardian have to be implemented in dedicated hardware. Furthermore the integration in a single chip improves reliability and decreases system costs. Due to ongoing protocol improvement activities a programmable solution for the communication controller was preferred. These requirements lead to an implementation with a programmable control unit, supported by several highly specialized units needed for an efficient implementation of the protocol.

3.2 Implementation

Figure 3 shows the block diagram of the TTP/C controller TTA-C2. The Protocol Control Unit (PCU) is the central control circuit. It coordinates the register transfer based communication between the functional units and executes high-level protocol tasks at 40 MHz-clock rate. Functional units are connected through the internal 16-bit wide data bus, which provides a common interface for these units. The following sections give a short overview of the provided functionality /7/.

3.2.1 Protocol Control Unit

The PCU is implemented as an application specific 16-bit instruction processor with three pipeline stages for instruction fetch (IF), instruction decode (DEC) and execute (EX). Stage IF contains the program counter, reads instructions from the instruction memory and passes it to the decode stage. The instruction decode stage generates the control signals for the data bus (to move data between functional units) and the execute stage, and branches are resolved. The third stage contains the ALU, which is able to perform integer addition/subtraction, a wide range of logical and bit manipulation operations and shift / rotate operations.

Due to optimizing the instruction set for protocol execution, an instruction memory size of 16kB is sufficient. This memory is split up into two areas. A 8kB ROM holds the boot code, various safety critical subroutines and the load routines for RAM and Flash. The fast 8kB instruction RAM is loaded at power-up either from the integrated flash or through the host interface. Special access resolution logic allows moving the instruction RAM content to the build-in CRC unit by normal move instructions executed from the same RAM, enabling a fast permanent firmware self test.

To support efficient protocol execution for the two communication channels, a doubled general-purpose register file supports fast task switching.

A hardware watchdog timer and some error signal traps monitor the controller operation putting the TTA-C2 controller into the fail silent state (means no bus activity) on unexpected events. Diagnosis can then be done via the host interface.

3.2.2 Flash Interface

The protocol firmware of the PCU is located in an integrated flash memory block (32kB) on chip. A dedicated "Flash Unit" controls erase/program/read operations initiated by the PCU and supports special functions to assure data integrity.

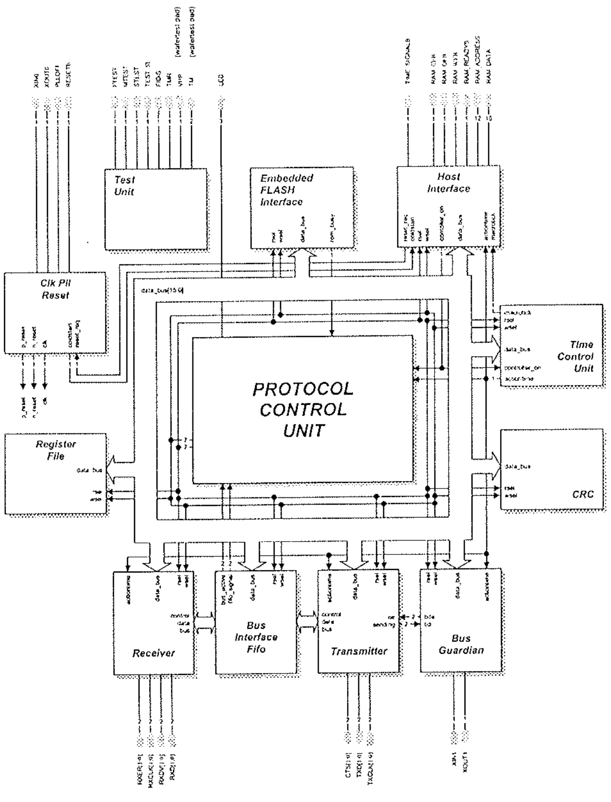


Figure 3: TTA-C2 Controller Architecture

3.2.3 Host Interface

The Host Interface unit implements the Communication Network Interface (CNI) between host subsystem and communication subsystem (Figure 1). It is implemented with a dual-ported RAM (4kB) which holds the messages exchanged among nodes. Also several status and control registers are accessible through this interface.

3.2.4 Time Control Unit

One key function of a time-triggered architecture is the generation of a global synchronized time base. The fault-tolerant average algorithm (FTA) is put in place for clock synchronization. The Time Control Unit enables an efficient implementation of the FTA. It consists of an adjustable counter, which allows fractional division of the system clock.

3.2.5 Transmitter, Receiver

Each controller contains a pair of independent receivers and transmitters with message fifo buffers to handle the physical layer of the two redundant serial communication channels between the TTP/C controllers. For each communication channel two different physical layer interfaces are provided, a low-speed interface for 5 Mbit/s to limit electromagnetic radiation and a high-speed interface for 25 Mbit/s usually using a fiber optics physical layer.

The receiver performs data synchronization and checks for noise, frame format and coding errors and watches the bus to check the temporal validity of transmitted frames.

3.2.6 CRC Unit

The CRC unit supports the calculation of cyclic redundancy checksums for two generator polynomials of 16 and 24 bits length. It allows the concurrent calculation of two checksums, one for each communication channel, in one clock cycle.

3.2.7 Bus Guardian

The bus guardian is an autonomous device that protects the channels from a timing failure of a controller. It contains a local crystal oscillator of its own in order to be able to tolerate a failure of the controller's clock. The bus guardian enforces the bus protection by applying plausibility and timing checks to the signals provided at the bus guardian interface.

3.2.8 Clock Pll, Reset

To meet industrial EMI requirements, the internal 40 MHz clock for the controller may be derived from an external 10 MHz crystal oscillator by a multiplying phase lock loop. Internal power-up reset generation and filter circuitry protects the flash memory contents and assures bus silence until proper startup of the controller.

3.2.9 Test Unit

Safety critical applications of the TTA-C2 controller demand test coverage of almost 100%. To shorten test time three test modes have been implemented: a flash test mode giving full access to the flash memory block; a functional test mode for memory test; and a scan chain mode for logic testing.

In order to test the RAM blocks we decided to make the instruction register accessible from outside. This makes it possible to pipe in arbitrary (memory) instructions and ob-serve the results on the data bus, which is connected to output pins in functional test mode. Since we can trigger all functions from outside we can also guarantee the basic functionality of each unit.

For the flash-memory-test x/y address, data, control lines and analog charge pump signals are observable to allow fast memory test and characterization during production.

Automated test pattern generation (ATPG) in conjunction with automated scan chain insertion is used for testing standard cell logic parts.

4. HDL Design Flow and Implementation Results

VHDL has been used to model the controller and insert the memory IP blocks. The controller design was functionally verified and synthesized to gate level including the various memory blocks. A complete top-down synthesis strategy was applied, i.e. the design was synthesized as a whole including scan-path insertion and automatic test pattern generation (ATPG). Special attention had to be paid on synchronizing the dataflow through various clock domains and integration of memory-IPs from different vendors.

The floor planning / placement / routing process not only had to consider timing constraints but also placement constraints (separated receiver/transmitter areas, bus guardian with own supply lines, dedicated transmitter-off circuits etc.) to support safety certification. Sign-off simulation with back annotated parameters for worst/best case analysis was done with a mixed language simulator.

All design steps and bug fixes during the design process had to be documented and presented to the customer as well as certification authorities during several design reviews to get approval for usage in X-by-wire applications.

Tabel 1 summarizes technical data and Figure 4 shows the chip layout.

5. Modeling Strategy and Verification

The modeling strategy focuses on hardware-software codevelopment and tool interoperability. It was our aim to do VLSI implementation and system validation in a single environment, i.e. to reuse the models for VLSI synthesis for system programming, functional tests for production test pattern generation, etc.

The RTL VHDL model of the communication controller is the interface for system simulation, verification and protocol programming, hiding the details of the hardware implementation, but the model also serves as reference for the synthesis process.

5.1 Chip Model

For the VLSI implementation the central VHDL model in the design environment is the register transfer level description. This description is restricted to a synthesizable subset of VHDL, all functional blocks are either synthesizable or can be mapped directly onto hardware IP blocks (e.g. RAM, Flash).

TTP/C protocol	Fully supported
Transmission speed	25 Mbit/sec MII
	5 Mbit/sec MFM
DPRAM to host	2kx16
Instruction ROM	4kx16
Instruction RAM	4kx16
GP Register	96x16
Flash memory	16kx16
Message fifo	2 x 19 words
CRC	16 / 24 bit
Clock frequency	max 40 Mhz
PLL clock	4 x XTAL clock
Analogue flash test	Build in interface
Technology	0.35μ CMOS
Die size	27 mm^2
Package	TQFP 80
Automotive Spec	OK

Table 1: Features and Technical Data

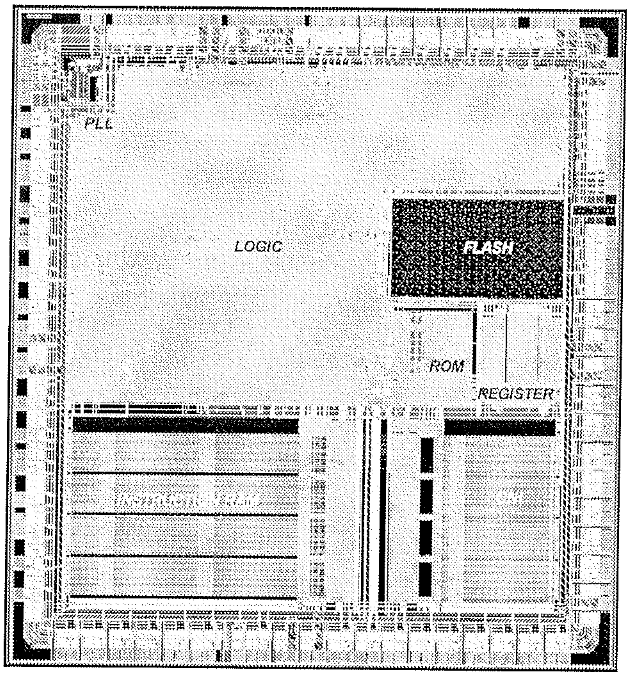
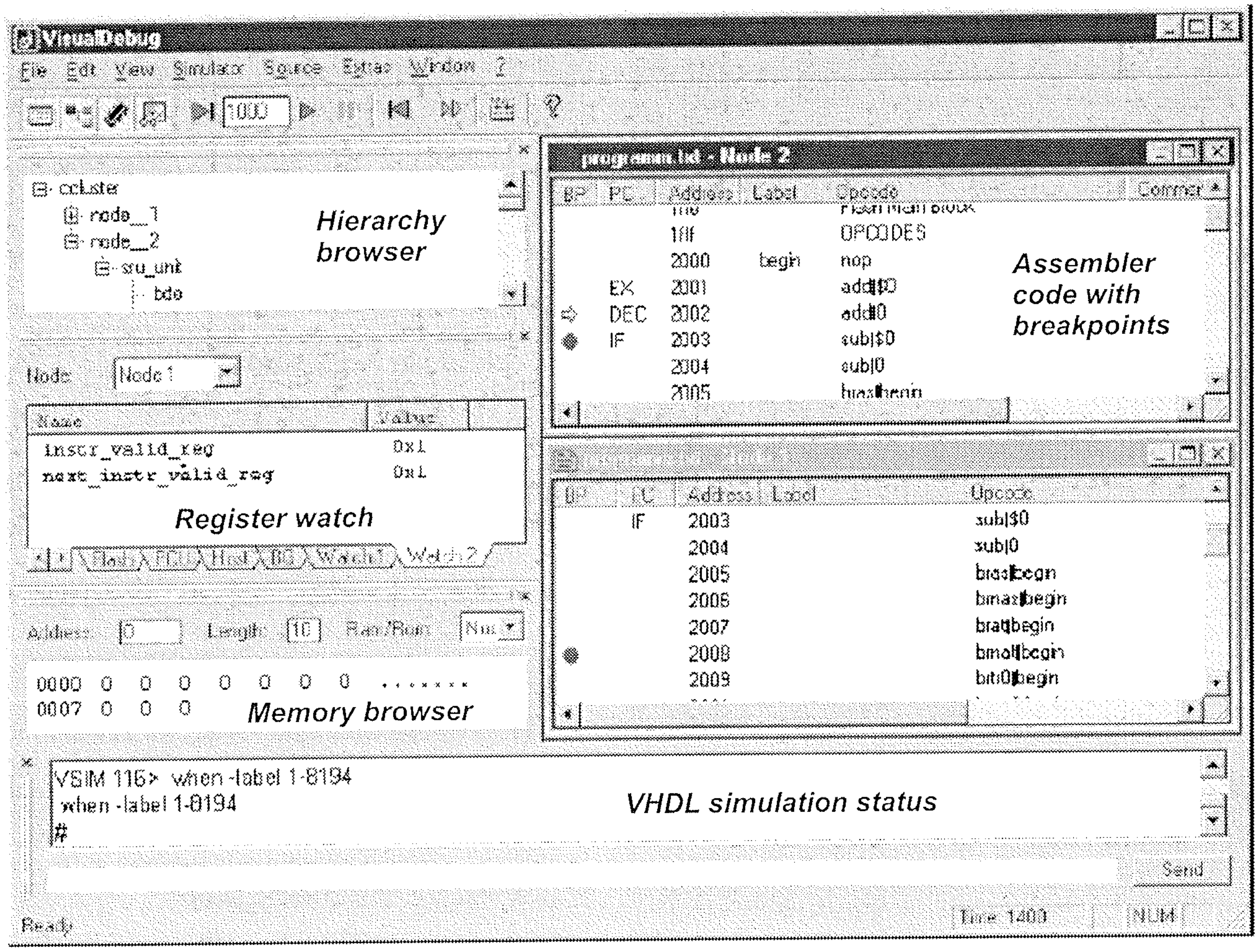


Figure 4: Layout View

5.2 Cluster Model

For the system level development simplified VHDL behavioral models of memory blocks, host controller card and cluster environment are provided to speed up simulation.



```
cclustr
  node__1
  node__2
    stu_unit
      bde
```

Name	Value
instr_valid_reg	0x1
next_instr_valid_reg	0x1

```
0000 0 0 0 0 0 0 0 .......
0007 0 0 0
```

```
BP  PC   Address  Label  Opcode
         1fff            OPCODES
         2000     begin  nop
    EX   2001            add|$0
⇨   DEC  2002            add|0
●   IF   2003            sub|$0
         2004            sub|0
         2005            bras|begin
```

```
BP  PC   Address  Label  Opcode
    IF   2003            sub|$0
         2004            sub|0
         2005            bras|begin
         2006            bmas|begin
         2007            brat|begin
●        2008            bmal|begin
         2009            bit0|begin
```

```
VSIM 116> when -label 1-8194
when -label 1-8194
#
```

Figure 5: Visual Debug Graphical Debugging Environment

5.3 HW-SW Development

To hide the complexity of the implementation from the protocol programmer a graphical debugging environment (VDEBUG, see Figure 5) and an assembler (TASM) were implemented. VDEBUG allows the protocol programmer to program the Protocol Control Unit and to verify firmware code of an entire cluster of controllers using the same controller model as used for VLSI implementation.

Of course the concept of using only one VHDL model slows down the system simulation environment to some degree compared to using a fast executing C model for protocol programming. But this drawback is more than compensated considering the effort and risk of keeping two models consistent during the whole design cycle. Additionally verification confidence is improved by stressing the same VHDL model code both from a hardware designers view by functional simulation pattern and a system programmers view by executing protocol firmware.

During the software development process macros and procedures, as well as block header and line comments have been extensively used. The resulting firmware has 4000 RISC assembler instructions (from 4096 possible) in total, whereas the ROM functionality has around 1000 RISC assembler instructions. The assembler source code with expanded macros has about 16000 LOC.

6. TTA Cluster and Node Design

In a distributed system the overall system functionality (e.g. brake-by-wire) is divided into several subsystems (e.g. pedal-sensing subsystem, brake-force calculation subsystem, and wheel-speed sensing subsystem). To feature composability and fault-tolerance, the interface between the subsystems needs to be specified in both the time- and value-domain.

6.1 Two-Level Design Approach

At system level, a system integrator (e.g. an automotive company) defines the subsystem functions and specifies the communication interfaces in the time- and value-domain precisely. At the subsystem level, the component supplier has to fulfill exactly these interface specifications but retains complete responsibility on all hardware and software design decisions to implement the desired functionality.

This Two-Level Design Approach /8//9/ is supported by a tool-chain (e.g. TTPtools), which allows the development and seamless integration of different subsystems into one distributed system.

In Figure 6, the upper half reflects the role of the system integrator. The cluster-design is the process of partitioning a system into several independent subsystems and defining the interfaces among each other. The result of the cluster-design process is a cluster-design database. The cluster-design process can be done using the tool TTPplan.

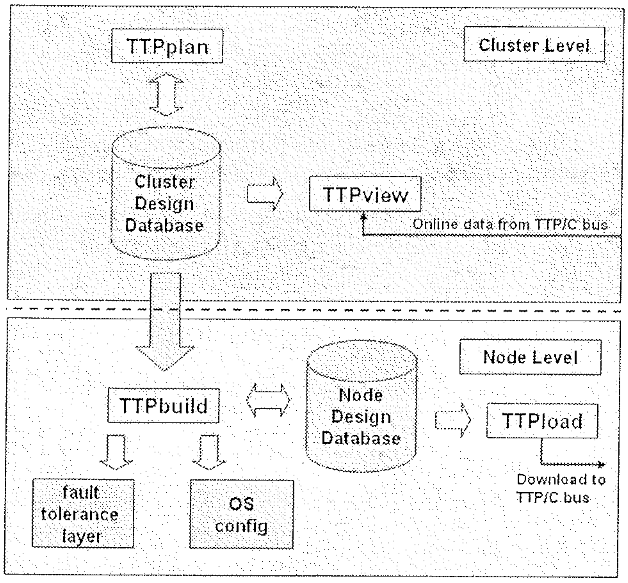


Figure 6: Cluster- and Node-Design

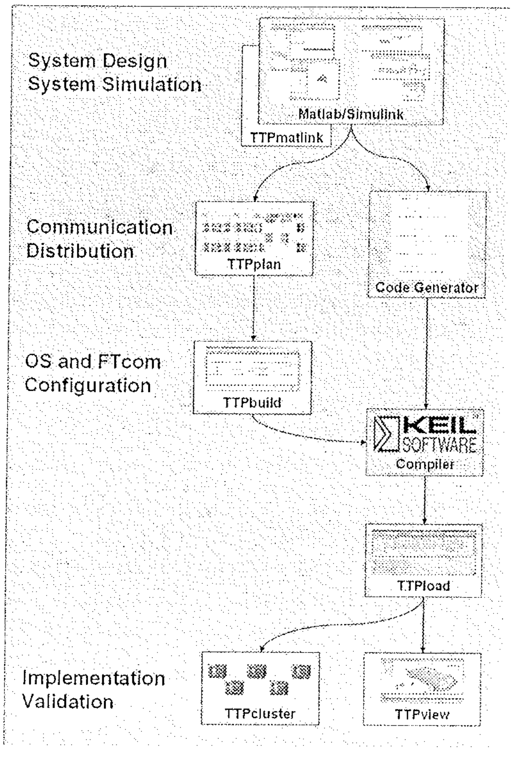


Figure 7: Model Based Design

The lower half of Figure 6 represents the role of the component supplier. At the node-design, individual subsystems are partitioned into software-tasks and the corresponding messages between each other are defined. The results of the node-design are the configuration information for the node's real-time operating system (e.g. TTPos), the automatic generated source-code for the fault tolerance layer (e.g. OSEKtime compliant FTcom layer) and a node-design database. This node-design process can be done using the tool TTPbuild.

6.2 Model Based Design

Figure 7 shows, how the TTPtools can be used in the case of a model based design approach. TTPmatlink is a Matlab/Simulink blockset, which enables a behavioral simulation of the distributed system in the time- and value-domain. It partitions the system into subsystems and it partitions these subsystems into tasks. Furthermore, it defines the messages between the subsystems and the tasks.

For the sake of completeness, the tool TTPload is used to download the communication schedule into the TTA-C2 controller's memory and to download the application files into the host controller's memory. The tool TTPview is used to monitor and debug the distributed system using a specific hardware (monitoring node), which is passively listening to all messages on the communication bus.

6.3 Example TTA Application

A Time-Triggered Car (TTcar) demonstrator (see Figure 8) has been implemented during the last two years at CTI. The TTcar is used for both educational purpose and to show drive-by-wire principles. It consists of 11 subsystems (one of them is triple redundant), 23 tasks and 11 nodes. The nodes are built up using the TTA-C2 communication controller and an Infineon C167CR (20 MHz) host microprocessor.

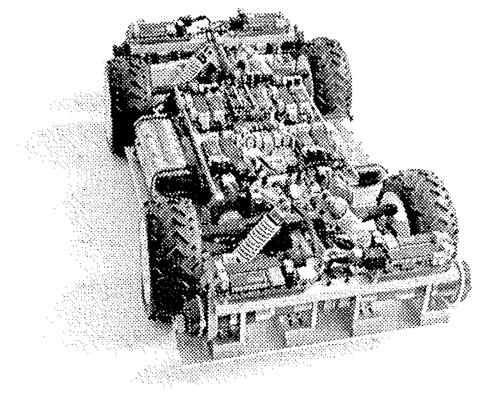


Figure 8: TTcar Application

Figure 9 shows *host1*, which is executing the *funk* and *drive_steer* subsystems. Each of these subsystems consists of two independent tasks. In Figure 10 the communication messages, which are exchanged between the t_funk_receive and the t_drive_steer tasks, are shown.

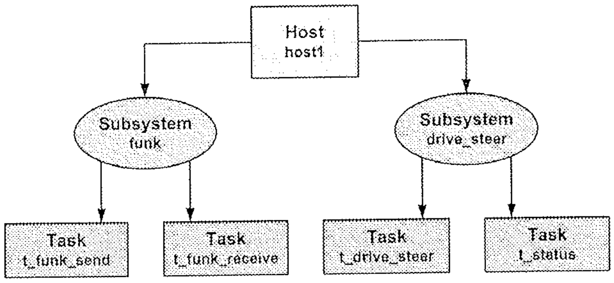


Figure 9: *Host, Subsystem, Task Relationship*

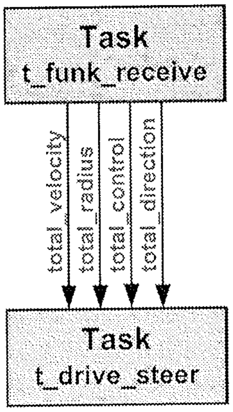


Figure 10: *Task and Messages*

7. Proof of concept and implementation

For the developers and users of TTA, the proof of concept and implementation is of utmost importance since its main application areas are safety critical and highly dependable systems like airplanes or cars. Requirements, like failure rates in the order of 10^{-9} failures per hour are beyond what can be proofed using methods like testing or simulation. According to this example, more than 100.000 years testing time would be needed. These circumstances lead to the increasing application of formal analysis and formal verification. Both rely on a formal model, which represents the systems properties. In the best case formal verification leads to general and complete statements about the properties of a system as such whereas testing and simulation leads to a probabilistic statement about its expected behavior (see Figure 11).

The combined usage of formal verification and experimental proof finally leads to a trustworthy statement about the reliability and more general dependability of safety critical systems build using the TTA.

7.1 Formal Proof of TTP

The clock synchronization algorithm used in TTA is a modification of the *Welch-Lynch* algorithm. The Welch-Lynch algorithm is characterized by use of the *fault-tolerant midpoint* as its averaging function. It tolerates a single arbitrary fault whenever equal to or more than four clocks are present. Minor formally verified the Welch-Lynch algorithm and Pfeifer, Schwier and von Henke formally verified its TTA instantiation.

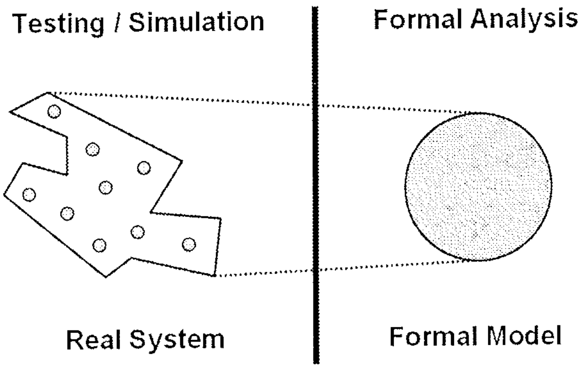


Figure 11: *Testing versus Formal Analysis*

In keeping with the *never give up* philosophy that is appropriate for safety-critical applications, TTA remains operational with less than four good clocks however the applied fault model changes. Rushby extended the fault model used for the formal verification to support not only arbitrary faults /10/.

The clock synchronization algorithm tolerates a single arbitrary fault only. Diagnosing the faulty node and reconfiguring to exclude it tolerate additional faults. The group membership algorithm of TTA, which ensures that each TTA node has a record of which nodes are currently participating correctly in the TTP/C protocol, performs this diagnosis and reconfiguration. Pfeifer formally verified the group membership algorithm in respect to its validity, agreement and self-diagnosis /11/.

The *transmission window timing* guarantees that messages sent by no-fault nodes do not collide on the bus. A faulty node, however, could broadcast at any time - it could even broadcast constantly. This fault is countered by use of a separate *fault containment unit* called *guardian* that has independent knowledge of the time and the schedule: a message sent by one node will reach others only if the guardian agrees that it is indeed scheduled for that time. The transmission windows timing algorithm has been formally verified by Rushby /12/.

7.2 Experimental Proof of TTP

The objectives of the European Commission (EC) sponsored project *Fault Injection for TTA* were to proof the concepts and implementations of the TTA by means of fault-injection experiments thoroughly. Six academic (Chalmers University, CTI, TU Pilsen, TU Prag, TU Valenzia, and TU Wien) and three industrial partners (Motorola, TTTech, and Volvo) have been carrying out experiments on different levels of abstraction (specification, chip, protocol, and node level) for more than two years /13/:

- **Protocol Microcode Fault Injection** (see Figure 12): Systematic injection of illegal instructions and instruction transformation of the TTP/C controllers protocol firmware
- **Heavy-Ion Fault Injection** (see Figure 13): A Californium-256 source causes internal and random faults in the CMOS structure of the TTP/C controller
- **Hardware Implemented Fault Injection**: Faults are injected into I/O pins of the TTP/C controller
- **C-Sim Fault Injection**: Independent C-language reference-implementation of the TTP/C protocol to check the consistency of the TTA specification
- **VHDL-based Fault Injection**: Injects faults into the VHDL-model of the TTP/C controller
- **Software Implemented Fault Injection**: Black-box and white-box fault injections into the code- and data-area of the TTP/C controller

The FIT project showed that there are still improvements, mainly at implementation level, possible and necessary. It also displayed that the main concepts and algorithms of TTA are correct and resist against brutal-force fault-injection experiments. As one of the most important consequences the change from a bus- to a star-topology, using a central guardian, needs to be mentioned. It does not only improve the error detection coverage drastically but it also separates the TTP/C controller and its guardian into two spatially separated fault containment units.

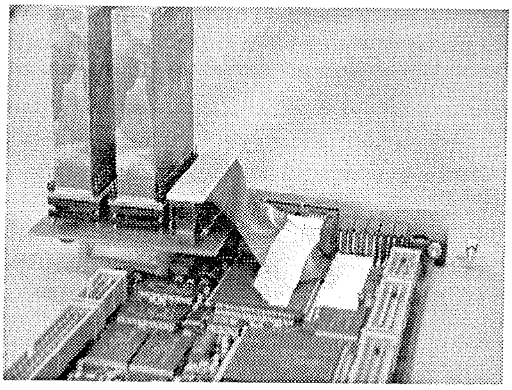


Figure 12: Protocol Microcode FI

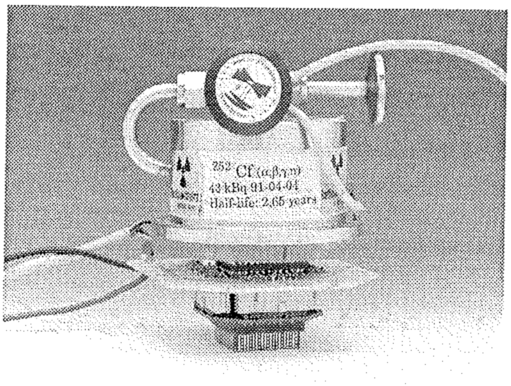


Figure 13: Heavy Ion FI

8. Conclusion

The Time Triggered Protocol (TTP/C) was implemented into the first industrial single-chip communication controller at Carinthia Tech Institute (CTI) in cooperation with austriamicrosystems, Unterpremstätten and TTTech, Vienna.

An application specific RISC core supported by highly specialized peripheral units was evaluated as well suited architecture to fulfill both cost and safety requirements. Design flow and implementation not only had to focus on functionality but also on system approval by safety authorities.

Furthermore, a complete simulation and programming environment useable for chip design as well as for system level design was developed. Related research projects were started in parallel to verify TTA principles and their implementation at all levels of a distributed hierarchical embedded system.

Consequently applying 'best practice' design techniques and combining industry experience with academic research spirit turned out to be a key factor of success in solving such complex design problems. The necessity to cover a widespread range of research subjects on one hand and the fact of limited budget and resources of academic institutions on the other hand inevitably leads to the setup of large multi partner projects. Whereas all partners benefit from such interdisciplinary collaborative work, the professional technical and financial management of such big projects is a new challenge to the academic community.

Resulting controller chips proved to be first time right and were nominated as 'Top Product of the Year' at 2002's SAE (Society of Automotive Engineers) world congress in Detroit. Beside other application fields, the TTA system received approval by US FAA (Federal Aviation Administration) and is going to be used for airplane engine control.

9. References

/1/ Scheidler Ch., Heiner G., Sasse R., Fuchs E., Kopetz H., Temple Ch. "Time-Triggered Architecture (TTA)". *Advances in Information Technologies: The Business Challenge,* IOS Press, pp. 758-765, 1998.

/2/ Kopetz H., Gruensteidl G. „TTP - A Protocol for Fault-Tolerant Real-Time Systems". *IEEE Computer,* 1994, Vol.: 24 (1), (pp. 14-23).

/3/ Kopetz H., Fuchs E., Hexel R., Krüger A., Krug M., Millinger D., Nossal R., Pallierer R., Poledna S., Schedl A., Sprachmann M., Steininger A., Temple Ch. "Specification of the basic TTP/C protocol". *Technical Report 18/96,* Institut für Technische Informatik, Technische Universität Wien, Vienna, Austria, Dec. 1996.

/4/ Sprachmann M., Grünbacher H. „TTA-C1: A Communication Controller Cell for Reuse". *Proceedings FDL'98 Forum on Design Languages,* Lausanne, Switzerland, 1998.

/5/ Sprachmann M. "Modeling a Controller for a Time-Triggered Protocol". *PhD thesis,* Vienna University of Technology, Jan. 1997.

/6/ http://www.ttpforum.org

/7/ Ley M., Grünbacher H. "TTA-C2, a SINGLE CHIP COMMUNICATION CONTROLLER for the TIME-TRIGGERED-PROTOCOL". *Proceedings ICCD 2002,* International Conference on Computer Design 2002, Freiburg, Germany

/8/ H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Kluwer Academic Publishers, 1997, ISBN 0-7923-9894-7

/9/ S. Poledna, H. Angelow, M. Gluck, I. Smaili, G. Stoger, C. Tanzer, TTP, "Two-Level Design Approach: Tool Support for Composable Fault-Tolerant, Real-Time Systems", *SAE 2000 World Congress*, 2000, Detroit, MI, USA

/10/ J. Rushby, "An Overview of Formal Verification For the TTA", *FTRTFT02*, Oldenburg, Germany, September 2002

/11/ H. Pfeifer, "Formal verification of TTA group membership algorithm", *Formal Description Techniques and Protocol Specification*, Pisa, Italy, October 2000

/12/ J. Rushby, "Formal verification of transmission window timing for the TTA", *Technical Report*, Computer Science Laboratory, SRI International, CA, March 2001

/13/ http://www.cti.ac.at/fit

Manfred Ley; Christian Madritsch
Fachhochschule Technikum Kärnten,
Carinthia Tech Institute
Europastraße 4, A-9500 Villach, Austria
m.ley@cti.ac.at, c.madritsch@cti.ac.at

Prispelo (Arrived): 15.09.2003 *Sprejeto (Accepted):03.10.2003*

CURRENT TRENDS IN EMBEDDED SYSTEM TEST

Franc Novak

Jozef Stefan Institute, Ljubljana, Slovenia

INVITED PAPER
MIDEM 2003 CONFERENCE
01.10.2003 - 03.10.2003, Grad Ptuj

Abstract: *Increasing complexity of electronic components and systems makes testing a challenging task. With the introduction of surface mounted devices, traditional in-circuit test techniques utilizing a bed-of-nails to make contact to each individual lead on a printed circuit board are becoming very costly and also inefficient. The need of an alternative test access fostered the development of novel test solutions like the IEEE 1149.1 boundary-scan architecture, recently extended to the mixed-signal test area by the IEEE 1149.4 Standard. At the chip level, technology advances allow to integrate functions that have been traditionally implemented on one or more complex printed circuit boards into one single integrated circuit. The development of such a System-on-Chip (SoC) is based on the design technique which integrates large reusable blocks (i.e., cores) that have been designed and verified in earlier applications in practice. This design technique introduces new extremely difficult test problems due to the fact that the core user (SoC designer) in most cases does not have detailed knowledge about the core design. Further difficulties represent the problem of test access of deeply embedded cores and portability of tests between core providers, SoC designers, as well as final SoC users. Embedded system testing faces all the above problems hence it is imperative to be aware of the novel test techniques and current trends in test.standardization. The paper briefly summarizes current state-of-the-art and gives pointers for further research in this topic.*

Sodobni pristopi k preizkušanju vgrajenih sistemov

Izvleček: *Zaradi naraščajoče kompleksnosti elektronskih komponent in sistemov postaja njihovo preizkušanje vedno večji problem. S površinsko montažo komponent postajajo tradicionalne metode preizkušanja osnovane na direktnem dostopu posameznih točk na tiskanini preko vzmetnih kontaktov drage in neučinkovite. Potreba po drugačni izvedbi dostopa do internih točk preizkušanca je vzpodbudila razvoj novih preizkusnih metod, kot je na primer IEEE 1149.1 (robna preizkusna linija) in njena razširitev na področje preizkušanja mešanih analogno-digitalnih vezij IEEE 1149.4. Tehnološki razvoj omogoča integracijo funkcij, ki so bile v preteklosti izvedene na enem ali več v modulih sistema, v enem samem integriranem vezju. Razvoj takšnega "sistema-v-čipu" (angl. System-on-Chip, SoC) je osnovan na načrtovalskih pristopih, ki omogočajo integracijo velikih že uporabljenih in v praksi preizkušenih logičnih blokov (jeder). Ta pristop pa hkrati prinaša tudi nove, izredno težke probleme pri preizkušanju načrtovanega produkta, saj razvijalec običajno ne pozna do podrobnosti zgradbe uporabljenih jeder. Nadaljnje težave predstavlja dostop globoko vgnezdenih jeder ter prenosljivost preizkusnih postopkov med dobavitelji jeder, načrtovalci sistemov-v-čipu in končnimi uporabniki. Preizkušanje vgrajenih sistemov se srečuje z vsemi navedenimi problemi, zato je pomembno, da so razvijalci seznanjeni s sodobnimi preizkusnimi metodami in njihovo standardizacijo. V članku na kratko povzemamo sedanje stanje in podajamo glavne smernice za nadaljnje poizvedbe na tem področju.*

1. Introduction

Advances in deep submicron technology are increasing the operating frequency and complexity of VLSI circuits which makes the testing problem more and more difficult. Complex Systems-on-Chip (SoCs) with the working frequency already in excess of 1 GHz require sophisticated testers with comparable clock rate. High-speed clocks employed in today's design require at-speed test to address potential performance-related problems. Current test systems are limited in signal generation and data capture speed to about 1.6 GHz and the cost of a tester capable of applying test vectors at the above clock rate approaches $10k per pin, not to count the additional cost of signal generators and measurement instruments needed for testing mixed-signal circuits. According to the SIA roadmap /1/, the prices of ATE (automatic test equipment) system are expected to continue toward > $20 million and may reach $50 million by 2010.

Another feature that impacts on test complexity is increasing transistor density of a chip which doubles every 18 to 24 months. This trend, known also as Moore's Law, continues to hold from the mid-1970s. Testing difficulty increases due to the fact that the number of transistors in a chip increases faster than the pin count. Consequently, internal chip modules become increasingly difficult to access. As described in /2/, the increase of test complexity can be expressed by the ratio N_t / N_p, where N_t denotes the number of transistors and N_p the number of input/output pins. The two parameters are related by the expression $N_p = K\sqrt{N_t}$,where K is a constant. This relation is known as Rent's rule. Since modern technologies allow drastic increase of transistor density in comparison with the number of pins, ATE systems have to access a larger number of complex logic blocks on a chip through a proportionally smaller number of input/output pins.

Due to the costly ATE systems, currently many factories around the word have installed test capability only at about 100 MHz clock rate which no longer fulfils the requirements of at-speed test of current circuit designs. Furthermore, the growing bandwidth gap as a result of the limited number of input/output pins prolongs the test time resulting in increased test cost.

Presented problems fostered development of new design-for-test (DFT) techniques with the goal of providing cost-effective high-quality at-speed test. In order to avoid the communication bottleneck between the high-performance ATE and the device-under-test (DUT), the *embedded test* approach /3/ implements ATE functions like high-speed generation of test vectors and response analysis together with some additional test control logic in the target DUT. In this approach, at-speed test actions are performed by the embedded test logic including pattern generation, test result compression and timing-generation. The remaining low-speed operations required for the execution of the complete test of the DUT are left to a low-cost external ATE, as shown in Figure 1. Alternatively, they can be completely integrated in the DUT in a built-in self-test (BIST) structure /4/.

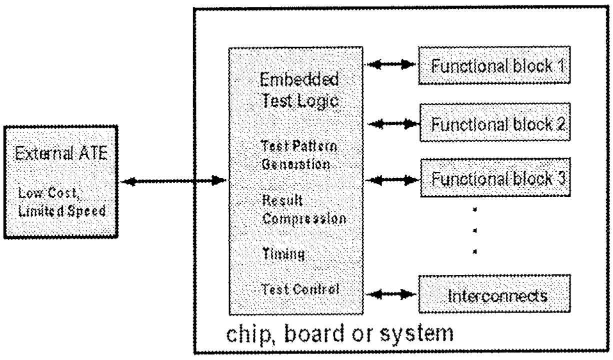


Figure 1: Embedded test configuration

Very high scale of integration of ICs introduces test problems also at other levels. Nowadays, both printed circuit board test and system test deal with very complex ICs which are by default difficult to stimulate. In addition, printed circuit boards are increasingly difficult to test by the conventional in-circuit test systems. Surface mounted devices placed on both sides of a board and smaller pin-to-pin spacing limit the access to the test points on the board, which makes the implementation of the bed-of-nails fixtures difficult and costly. The need for an alternative access of internal test points gave the idea of building the test probes directly into the chips and to connect the probes with the external ATE by simple serial line. The effort of ATE manufacturers and EDA tool suppliers organized as the Joint Test Action Group (JTAG) resulted in a *boundary-scan* test technique for digital circuits and systems and was approved as the IEEE Standard 1149.1 in 1990 /5/. The main objective was to provide standardized approaches to board interconnect test and internal test of the devices placed on it. As the standard gained popularity in practice, many other applications in test as well as in other areas have been reported. In 1999, IEEE Standard 1149.4 (Standard for a Mixed-Signal Test Bus) /6/ has been approved. This standard can be regarded as the extension of the IEEE Standard 1149.1 to the area of mixed-signal device test.

Let us finally mention the emerging IEEE P1500 Standard for Embedded Core Test /7/ which offers standardized DFT solutions for SoCs but will affect also board and system test once the chips complying to the IEEE Standard P1500 appear in practice.

2. DFT Standards

Many embedded test solutions in practice rely on the test infrastructure imposed by the above-mentioned standards. Standard test port features defined by these standards provide standard way to deliver test data to the DUT and capture test results. They also facilitate built-in self-test (BIST) implementation and provide effective means to reuse BIST and embedded test solutions at the board and system level test. For this reason we shall in the following briefly review the main features of IEEE Standards 1149.1, 1149.4 and P1500.

2.1 Boundary Scan Standard

The basic idea of the boundary-scan approach is to replace external probe or bed-of-nails pins by internal probes placed on a chip between device pins and IC core logic. During normal operation, the internal probes (i.e., scan modules) are transparent while in the test mode additional test logic is activated between the external pin and IC core in order to perform the selected test. Scan modules are serially interconnected and form a boundary-scan register around the chip. The boundary-scan register allows the application of test vectors to the IC's pins or core circuit as well as sampling of internal and external values at IC pins. Test vectors/responses are serially shifted in and out of the boundary-scan register through the device's test ac-

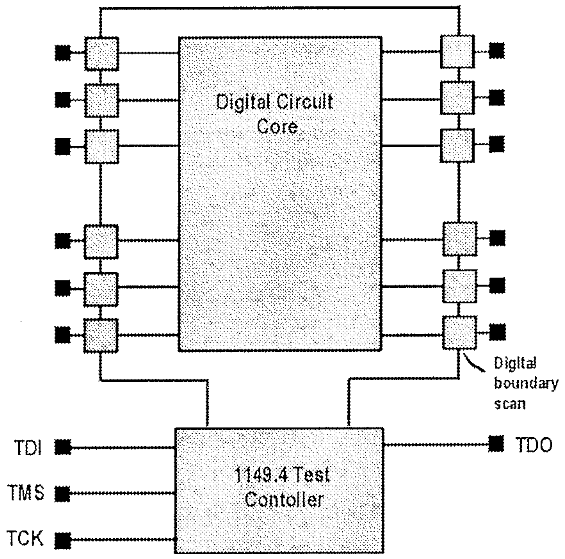


Figure 2: A schematic of a chip complying to IEEE Standard 1149.1

cess port (TAP). TAP consists of four mandatory connections: Test Clock input (TCK), Test Mode Select input (TMS), Test Data Input (TDI) and Test Data Output (TDO) and optional Test Reset input. A schematic of a chip with included IEEE Standard 1149.1 test infrastructure is shown in Figure 2. In a typical application, chips on a board are configured into one or more boundary-scan chains with common TCK and TMS connected to the external test system.

A chip compliant with IEEE Standard 1149.1 includes TAP, TAP controller and at least the following three test data registers: Boundary-Scan Register, Instruction Register and Bypass Register. TAP Controller generates control signals required for the operation of the Instruction Register and the Test Data Registers. IEEE Standard 1149.1 prescribes three mandatory instructions: Sample/Preload, Bypass and Extest. The Sample/Preload instruction is used to obtain a snapshot of the device input and output signals during the normal operation. Its execution does not interfere with circuit's normal operation, hence this option may be very useful for system debugging. The purpose of the Bypass instruction is to shorten the scan path through the boundary-scan architecture when scan access of the test data registers is not required. The Extest instruction allows test of board interconnections (i.e., test of opens, shorts or bridging faults). It can be also used to test non-boundary-scannable parts of the system. A chip compliant with IEEE Standard 1149.1 may optionally include other types of test data registers to perform non-mandatory instructions like, for example, Runbist (which runs a built-in self-test of a circuit), Idcode (which allows reading of the circuit's identification code and thus permits blind interrogation of the assembled components on a board), Intest (which allows slow speed testing of the core of a circuit), and many others.

Conventional boundary-scan tests run at the test clock TCK generated by a low cost external test system. Such configuration is primarily used for static interconnect test. However, as we shall see in the next section, one of the reported embedded test solutions extends boundary-scan test infrastructure to perform at-speed test.

2.2 Mixed-Signal Test Bus Standard

IEEE Standard 1149.4 defines the way to access the mixed-signal chips on a board in order to perform interconnect test and parametric test of discrete components connected to the chips. IEEE Standard 1149.4 provides facilities that allow to detect opens in the interconnections between chips, and to detect and localize bridging faults. The defined test infrastructure allows interconnect testing in full compatibility with IEEE Standard 1149.1. It also allows measurements of the values of discrete components such as pull-up resistors, filter capacitors, etc., that are often interposed between integrated circuits on a board. In addition, facilities to perform internal test of a mixed-signal chip can be provided. This option is not mandatory.

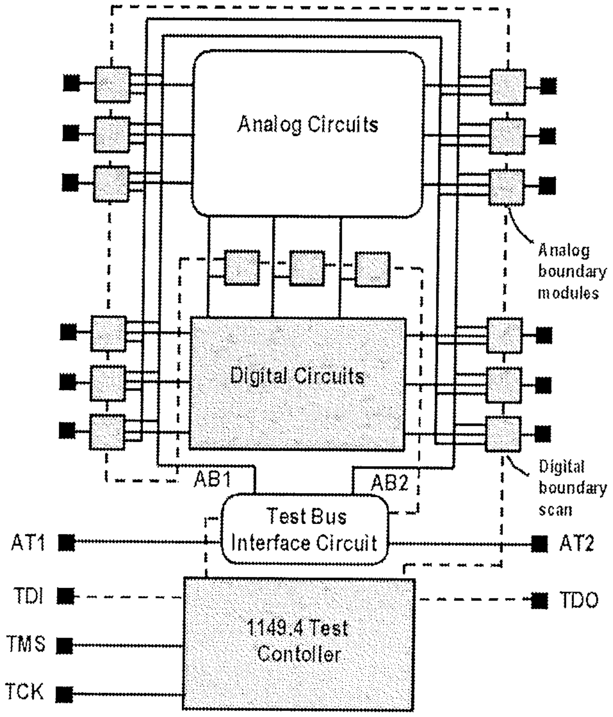


Figure 3: A schematic of a chip complying to IEEE Standard 1149.4

IEEE Standard 1149.4 can be regarded as an extension of IEEE Standard 1149.1. The 1149.4 extensions are analog boundary modules (ABMs) on analog functional pins accessed via internal analog test bus (AB1, AB2). The bus is connected to the Analog Test Access Port (ATAP) through the Test Bus Interface Circuit (TBIC). Digital pins have boundary cells as specified in IEEE Standard 1149.1. A schematic of a chip with included IEEE Standard 1149.4 test infrastructure is shown in Figure 3.

Four mandatory instructions are prescribed: Sample/Preload, Bypass and Extest (since the digital part is compliant with IEEE Standard 1149.1) and Probe. The Probe instruction allows analog pins to be monitored on the analog bus AB2, and/or stimulated from the analog bus AB1 while the chip is operating in its normal operation state.

2.3 Standard for Embedded Core Test

SoC design integrates large reusable blocks (i.e. cores) that have been designed and verified in earlier applications in practice. Embedded cores provide a wide range of functions, like CPUs, DSPs, interfaces, controllers, memories, and others. The cores put together in a SoC normally originate from different core providers. In order to protect their intellectual property core providers do not completely reveal design and implementation details which makes the problem of SoC testing rather challenging to the core user (i.e., SoC designer). On the other hand, correct operation of a core in the target SoC is of interest of both core user and core provider. In order to provide an independent openly defined design-for-testability method for integrated circuits containing embedded cores, an initiative to develop a standard has been taken by the IEEE P1500 Working Group, /7/.

The main entity of the test architecture defined by IEEE Standard P1500 is a *test wrapper* placed around each core of a SoC. Test wrapper provides interface between the embedded core and its environment. For testing a core, a test source generating test vectors and a test sink collecting the test responses must be provided. Test access mechanism (TAM) transports test vectors from the source to the core and test responses from the core to the test sink. It also allows testing of interconnects between SoC cores. Standard prescribes mandatory serial TAM and allows optional user-defined parallel TAMs.

The test wrapper, shown in Figure 4, connects the terminals of the core to the rest of SoC during the normal operation and to the test access mechanism in the test mode. Wrapper operation is controlled by a set of control and clock signals provided at the Wrapper Interface Port (WIP). WIP also includes Wrapper Serial Input (WSI) and Wrapper Serial Output (WSO) which are used to shift-in and shift-out serial test data. Test wrapper contains the following mandatory registers:

- wrapper instruction register (WIR) which is similar to IEEE 1149.1 instruction register and controls the operation of the wrapper. WIR receives instructions via wrapper serial input WSI.
- wrapper boundary register (WBR) to which the core functional terminals are connected. It is a serial shift register similar to the IEEE 1149.1 boundary-scan register.
- bypass register which is similar to IEEE 1149.1 bypass register. It is used to bypass the WBR. In a single scan path configuration, bypass registers enable to skip out the WBRs of the cores that are not being tested.

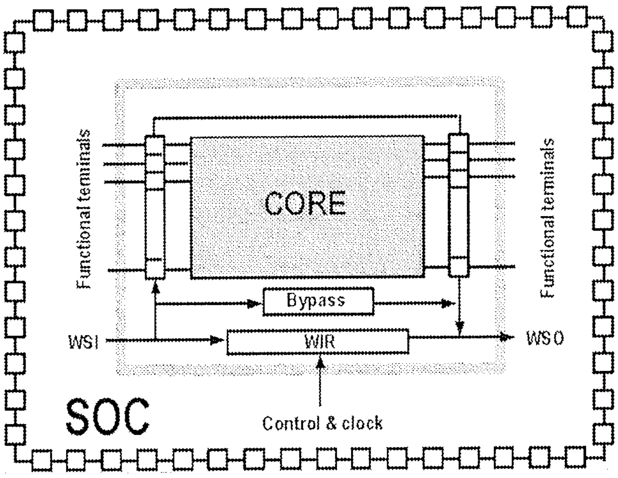


Figure 4: IEEE Standard 1500 test wrapper structure

Core-internal and core-external tests can be performed. Core-internal test is based on the test information that the core user gets from the core provider. It may consist of the application of test patterns within a specified test protocol, or of initiation of a built-in self-test of the core. Core-external test checks external connections between the cores and additional glue logic designed by the SoC integrator.

For the core-internal tests, test stimuli are provided via TAM to wrapper boundary at the core input terminals and test results are read via TAM from the wrapper boundary at the core output terminals. For the core-external tests, initial logical values are set-up via TAM at the wrapper boundary at the core output terminals and results are observed at the wrapper boundary at the core input terminals.

IEEE Standard P1500 prescribes three mandatory instructions: WS_BYPASS (which places wrapper bypass register between the WSI and WSO of the wrapper), WS_EXTEST (which allows testing of off-core circuitry and interconnections between cores) and Wx_INTEST (a user specified core test instruction).

2.4 Applications in practice

IEEE Standard 1149.1 is widely accepted standard supported by semiconductor industry and EDA tool providers. Since its introduction, the availability of devices conforming to the standard has increased steadily. Boundary-scan is an efficient technique for detecting and localizing manufacturing faults such as shorts, opens and component misplacements. The same test infrastructure is used to support built-in self-test (BIST) capabilities of complex devices. Extensive number of papers and technical reports are available on the web. For the newcomers, introductory book on boundary-scan /8/ may prove advantageous. Another major application of the 1149.1 boundary-scan infrastructure is the so-called In-System Configuration (ISC). ISC is the ability to load configuration data into a programmable device, such as a CPLD or a FPGA, via boundary-scan path. Standardization efforts in this area resulted in the IEEE 1532 In-System Configuration of Programmable Devices Standard /9/, approved in late 2000.

While the use of IEEE 1149.1 has become a prevalent solution for board and system manufacturing test, its analog counterpart IEEE 1149.4 still lacks the support of major electronic manufacturers. The absence of IEEE 1149.4 compatible devices prevents the designers to include a standardized mixed-signal test structure into their systems. So far, only a few experimental test chips supporting the standard have been reported /10/, /11/, /12/. On the other hand, the standard attracted the attention in research and academia. Beside the introductory book /13/, different test and measurement methods using IEEE 1149.4 test infrastructure have been proposed /14/ - /17/.

IEEE Standard P1500 is about to enter ballot process in the following months hence solutions in compliance with the standard could not yet be reported. But according to the expressed interest of core providers and SoC designers we can expect that the standard will gain wide support in practice. Recently, numerous papers have been published on P1500 related issues in the proceedings of the International Test Conferences and the DATE Conferences. Papers /18/ - /22/ can serve as starting points for further reading on this subject.

Described standards are not stand-alone solutions. They are combined together with local DFT solutions to provide an efficient test. The IEEE Standard P1500 test wrapper, for example, provides means to perform internal core test via the scan chains originally implemented in the given core.

Application of DFT standards in practice is, of course, subject to a thorough economic analysis. However, when assessing the trade-off of a given DFT solution, testing should be regarded primarily as a cost-avoidance strategy. The well-known "Rule of Ten" indicates that the cost to locate a fault increases about ten times at each subsequent testing stage. A DFT solution that facilitates the location of faults at earlier testing stages reduces the product cost. The author and his research group share the experience that even a simple awareness of the DFT principles contributes to the system's testability as well as dependability features /23/.

3. Advanced embedded test approaches

The basic principle of embedded test is the generation of test stimuli and analysis of test results on the unit-under-test instead of using for this purpose an external ATE. Generation of test stimuli and collection of test results is specific to the type of the functional block under test.

For testing digital logic, pseudo-random pattern generator (PRPG) as stimuli generator and multiple-input signature register (MISR) for collection and compression of test results are normally employed. The pioneering work on signature analysis technique of R.A.Frohwerk /24/ has been followed by a vast number of papers exploring theoretical limits of pattern generation and compaction by this technique as well as different possibilities of its application in practice.

Embedded memories represent another type of functional block that requires specific test solutions. Conventional ad-hoc methods use either additional logic to route the embedded memory inputs and outputs to the external pins of the chip or place a scan chain around the embedded memory for shifting in and out the test patterns. The first approach is not adequate due to extensive routing of extra interconnects and to the restrictions in pin count of the chip while the second exhibits prohibitive test time. An alternative way is memory BIST approach /25/ with on-chip (or on-board) generation of test patterns and compression of test results. The requirement of generating deterministic sequences of test patterns (i.e., marching test pattern) for testing target memory structures resulted in different test algorithms. Specific features of test problems and solutions have made embedded memory test a unique research area.

Mixed-signal functional blocks are another group of system parts that require different embedded test approaches. This heterogeneous group calls for specific measurement-based solutions completely different from those described so far. A concise introduction on this subject is the book written by M.Burns and G.W.Roberts /26/. Any further discussion on this topic is, however, beyond the scope of this paper.

For any embedded test solution, either at chip or at board level, it is advantageous to provide test data input/output via standardized test port. In this way, communication with external tester or implementation of system test is considerably simplified. In addition, implementation of local embedded test solutions in the frame of standard test infrastructure complying to previously described DFT standards allows portability and reuse of embedded tests at higher system levels.

As mentioned before, IEEE Standard 1149.1 and 1149.4 test infrastructure originally supports low speed interconnect test. However, with miner modifications of boundary-scan cells and small additional control logic it is possible to perform at-speed interconnect test /3/. Besides, novel techniques that exploit IEEE Standard 1149.4 test structures in high frequency measurements have been recently reported /12/, /27/, /28/.

4. Conclusion

Conventional test techniques are inadequate for testing complex modern SoCs, boards and systems due to the low bandwidth and low test speed, limited access of internal test points resulting in increased test time and test cost. Alternative approaches with embedded test solutions provide several advantages including cost-effective at-speed test, increased fault coverage and reuse of device test at the board or system level. In practice, most current test solutions rely on DFT standards. Basic knowledge of their principles and possible use of their test infrastructure is imperative for modern chip design. The paper gives a brief overview of the above issues and offers a list of related references.

5. References

/1/ International Technology Roadmap for Semiconductors, 1999 edition, Test and Test Equipment.

/2/ M.L.Bushnell, V.D.Agrawal, *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*, Kluwer Academic Publishers, 2000, ISBN 0-7923-7991-8.

/3/ B. Nadeau - Dostie, editor, *Design for at-speed test, diagnosis and measurement*, Kluwer Academic Publishers, 2000, ISBN 0-7923-8669-8.

/4/ C.E.Stroud, *A designer's guide to built-in self-test*, Kluwer Academic Publishers, 2002, ISBN 1-4020-7050-0.

/5/ IEEE Std 1149.1-1990: IEEE Standard Test Access Port and Boundary-Scan Architecture.

/6/ IEEE Std 1149.4-1999: IEEE Standard For a Mixed-Signal Test Bus.

/7/ IEEE P1500 Standard for Embedded Core Test: (working group web page) http://www.manta.ieee.org/groups/1500/

/8/ H.Bleeker, P. van den Eijnden, F. de Jong, *Boundary-ScanTest, A Practical Approach,* Kluwer Academic Publishers, 1993, ISBN 0-7923-9296-5.

/9/ IEEE Standard for In-System Configuration of Programmable Devices, IEEE Standard 1532-2000, IEEE, 2000.

/10/ "JTAG Analog Extension Test Chip: Target Specification for the IEEE P1149.4 Working Group," Preliminary Review 012, Keith Lofstrom Integrated Circuits, Beaverton, Ore., 1998; http://grouper.ieee.org/groups/1149/4/kl1p.html.

/11/ U.Kac, F.Novak, F.Azais, P.Nouet, M.Renovell, "Extending IEEE Std. 1149.4 Analog Boundary Modules to Enhance Mixed-Signal Test", *IEEE Design & Test of Computers*, Vol. 20, No. 2, March-April 2003, pp. 32-39.

/12/ S.Sunter, K.Filliter, J.Woo, P.McHugh, "A general purpose 1149.4 IC with HF analog test capabilities", Proc. Int'l Test Conf. (ITC 2001), IEEE Press, 2001, pp. 38-45.

/13/ A.Osserian, editor, "*Analog and Mixed-Signal Boundary Scan, A Guide to the IEEE 1149.4 Test Standard*", Boston, Kluwer Academic Publishers, 1999.

/14/ K.P.Parker, J.E.McDermit, and S.Oresjo, "Structure and Metrology for an Analog Testability Bus," Proc. Int'l Test Conf. (ITC 93), IEEE Press, 1993, pp. 309-317.

/15/ K. Lofstrom, "Early Capture for Boundary Scan Timing Measurements," Proc. Int'l Test Conf. (ITC 96), IEEE Press, 1996, pp. 417-422.

/16/ U.Kac, F.Novak, S.Macek, M. Santo Zarnik, "Alternative Test Methods Using IEEE 1149.4," Proc. Design, Automation, and Test in Europe (DATE 2000), IEEE CS Press, 2000, pp. 463-467.

/17/ S. Sunter and B. Nadeau-Dostie, "Complete, Contactless I/O Testing--Reaching the Boundary in Minimizing Digital IC Testing Cost," Proc. Int'l Test Conf. (ITC 2002), IEEE Press, 2002, pp. 446-455.

/18/ R.K.Gupta, Y.Zorian, "Introducing core-based system design", IEEE Design and Test of Computers, Vol. 14, No. 4, October-December 1997, pp. 15-25.

/19/ Y.Zorian, "Test requirements for embedded core-based systems and IEEE P1500", Proc. International Test Conference, Washington, DC, November 1-6, 1997, pp. 191-199.

/20/ Y.Zorian, E.J.Marinissen, S.Dey, "Testing embedded-core based system chips", Proc. International Test Conference, Washington, DC, October 18-23, 1998, pp. 130-143.

/21/ E.J.Marinissen, Y.Zorian, R.Kapur, T.Taylor, L.Whetsel, "Towards a standard for embedded core test: an example", Proc. International Test Conference, Atlantic City, NJ, September 28-30, 1999, paper 24.1.

/22/ E.J.Marinissen, R.Kapur, Y.Zorian, "On using IEEE P1500 SECT for test plug-n-play", Proc. International Test Conference, Atlantic City, NJ, October 1-5, 2000, paper 24.1.

/23/ B. Korousic Seljak, "Efficient task scheduling approach relevant to the hardware/software co-design of embedded system", *CIT. J. Comput. Inf. Technol.*, 2000, vol. 8, pp. 197-206.

/24/ R.A.Frohwerk, "Signature analysis: A new digital field service method", *Hewlett-Packard Journal*, Vol. 28, no. 9, May 1977, pp. 2-8.

/25/ S.K.Jain, C.E.Stroud, "Built-in self-testing of embedded memories", *IEEE Design & Test of Computers*, Vol. 3, No. 5, October 1986, pp. 27-37.

/26/ M.Burns, G.W.Roberts, "*An introduction to mixed-signal IC test and measurement*" Oxford University Press, 2000.

/27/ K.Lofstrom, "Early capture for boundary scan timing measurements", Proc. International Test Conference, Washington, D.C., October, 1996, paper 15.3.

/28/ J. Hakkinen, P.Syri, M.Moilanen, "A Frequency Mixing and Subsampling Based RF-measurement Apparatus for IEEE 1149.4", Proc. Board Test Workshop BTW03, Charlotte, October 2003.

Franc Novak
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
franc.novak@ijs.s

Prispelo (Arrived): 15.09.2003 *Sprejeto (Accepted): 03.10.2003*

EFFICIENT DEVELOPMENT OF HIGH QUALITY SOFTWARE FOR EMBEDDED SYSTEMS

Stanislav Gruden

Iskraemeco d.d., Kranj, Slovenia

INVITED PAPER
MIDEM 2003 CONFERENCE
01.10.2003 - 03.10.2003, Grad Ptuj

Abstract: *New electronics products are being developed with a constantly growing pace today. The development must meet very tough criteria: short time-to-market, continuous use of currently the best available technology in order to reach high performance requirements, etc. More and more we see that the cost for this is a decreasing quality of the products, especially the low cost consumer electronics. The problems are most often due to the insufficiently tested software of the embedded systems used. On the other hand there is no need to make the software optimized for performance anymore, usually the more efficient way of optimizing the overall cost and resources is just to use more powerful hardware.*

In order to increase the software quality in these hard development conditions some measures have to be taken into consideration: rigorous testing is one of them, but a lot can also be achieved by using of high level programming languages wherever applicable, making code as portable and reusable as possible, using tested other party software whenever accessible, etc.

Some techniques that can be used to make software more portable and reusable are presented. A common characteristic of these techniques is they use some of the system resources like memory or CPU time in exchange for structural organization that makes the code much easier to maintain, distribute between many developers and test. The technique of compiling and testing the code on strong personal computers or workstations before using it on a real system is described. This technique takes up some additional development resources at the beginning but saves them lately because it makes developing and testing a new code much easier, makes it portable and it is possible to have a large part of application completed even before the actual hardware is obtained, etc.

Učinkovit razvoj programske opreme za vgrajene sisteme

Izvleček: Živimo v svetu, kjer elektronske naprave razvijajo z vedno hitrejšim tempom. Ta razvoj se odvija v težkih pogojih: vstop na tržišče mora biti hiter, hkrati je potrebno slediti napredku na področju tehnologije z namenom ves čas delovati v optimalnem področju. Vedno bolj je očitno, da to gre na račun kvalitete izdelkov, posebno to velja za nizkocenovne širokopotrošniške naprave. Najpogosteje problemi nastanejo zaradi nezadostno preverjene programske opreme vgrajenih sistemov. Po drugi strani se izkaže, da ni več potrebe po pretirani optimizaciji programske kode, ampak je za boljšo izkoriščenost razvojnih virov in manjše skupne stroške boljše vzeti zmogljivejšo strojno opremo.

Za povečanje kakovosti programske opreme v teh zaostrenih pogojih so potrebni določeni ukrepi. Natačno testiranje je najpomembnejše. Mnogo se da doseči z uporabo višjih programskih jezikov, kjer je to mogoče, ter z izdelavo čimbolj prenosne programske kode in uporabo preverjene programske opreme drugih proizvajalcev.

Prispevek prikazuje nekaj načinov, kako programsko kodo narediti dobro prenosljivo in primerno za ponovno uporabo. Splošna značilnost takih metod je, da boljša strukturiranost programa gre nekoliko na škodo porabljenih virov - pomnilnika, procesorskega časa. Končni rezultat je lažje vzdrževanje kode, preprostejši prenos med razvijalci in učinkovitejše testiranje. Opisali smo postopek, kako na močnih osebnih računalnikih ali delovnih postajah prevesti in preveriti programsko kodo brez uporabe končne strojne opreme. Postopek zahteva dodatne razvojne vire na začetku, ki pa jih več kot prihranimo kasneje, saj sta razvoj in testiranje nove kode mnogo preprostejša, tako dobljena koda je sama po sebi dobro prenosljiva, večino kode lahko dokončamo tudi v primeru, da končne strojne opreme še nismo dobili, itd.

1. Introduction

Many articles and books have been written on the software crisis that has been continuously happening since the first commercial computer applications /1/. They address the huge problem of software quality, late time-to-market, etc.

The reasoning in this article is based on real life situations, which differ greatly from the theoretical situations. Theoretically the less expensive development would consist of a thorough problem analysis and after that, a complete design of software structure. The actual coding would not start until these two phases are finished and confirmed by the customers and design team. Unfortunately, this almost never is a case. The system analysis would usually take too much effort if it is to be complete and all the facts about the system are taken into account. This would also require the customer to really study the analysis and to make all necessary comments on time. Many customers are just not prepared to do this and leave the important decisions to the developers. The most frequent reason for omitting the analysis steps are the deadlines. We just simply live in a world where time is money and the 'theoretically less expensive development' would actually cost more because of the lost opportunities on the market /4/.

There exist possibilities, despite the real world requirements, to build up a better structured, more documented, portable and less error-prone code. Early and frequent integration /2/ does so with so-called front-loading - the problems should be detected as soon as possible. Designers must meet earlier and resolve the conflicts earlier. Following some guidelines ensures this to happen virtually automatically while doing the 'preferred' work - the coding.

Thorough testing must be performed through the whole development process. Some possibilities of how to set up and use testing features are presented.

The presented ideas are directly applicable when using standard programming languages like C, C++. They may not always be useful if higher-level design tools, which automate code generation, are used since these tools may already force the way the program is designed.

2. Relation between development resources, time to market, reliability and product price

The basic requirements for any development are:

- The final product must have a quality that is expected by the customers /2/. This may differ largely from one application to another. Mass consumer products like cheap children toys need not to be very reliable, sometimes it is even expected that they will be in use for some days and then abandoned; on the other hand the professional equipment is supposed to work without any problems for a long period of time. The highest level of reliability must be assured when human lives depend on the proper operation of the system.
- The production costs of the product tend to be minimized although the effort of doing this depends on application type. Very large quantity products are hardware minimized to the highest possible limit, since every cent is important. The software and hardware development costs for such products are low and represent a negligible part of the final price. On the other side high quality products need a lot of intensive development. Being sold in much smaller quantities means that every device incorporates in its price a significant portion of the development costs. These costs may be much higher than the hardware costs and in this case hardware cost minimization is not so important.
- Development resources are always limited. The most important of these resources are developers. Additional money, development tools and equipment can be obtained one way or another in case of time pressure, but human resources are not trivial to add. This fact was known long ago, as early as in year 1975 /1/. A lot of time is needed to find people that are able to do the work; their training and getting to know the project cause another delay. This is true even with experienced developers. In case of tough deadlines one usually does not have any other choice but to count only on the available developers. The exceptions are possible in a case a very distinctive module can be separated from the entire product, which has a simple application programming interface (API), easily describable black box functionality, and is at the same time so complex that it takes a lot of work to implement. The functionality and requirements for such a module can be easily documented and presented to a new, skilled developer.
- Time to market is a big issue /3, 4/. In the world where the market competition is the main driving force, it is absolutely necessary to get to the market as soon as possible. At the beginning the demand for a new product is at the strongest, there is less or no competition and the prices are high. When the competition comes to the market the prices may fall to production cost, development may not be covered anymore and the profit disappears. The applications also have short lives. In some cases a fast development also decreases development costs /3/.

The goal is to make a good compromise between these issues of which each one generally contradicts the other. Lower product price means weaker hardware, which needs more development resources to get the work done, otherwise the reliability will be worse or development time will be unacceptable high. As a very simple approximation it could be stated that a weighted sum of these parameters is a constant value.

The use of better tools and equipment can help reducing this value but may not always be affordable. On the other hand, a better organization of the development process can greatly improve the final quality of the product - the number and the severeness of operation errors (bugs), which are an inevitable part of the products, especially complex ones.

For quality devices it usually turns out that it is better to use more advanced platform than the minimal one (more memory, higher speed). Namely, despite good planning and previous analysis the amount of system resources on the platform needed is usually underestimated. With stronger platform the programming is easier because the focus can be put to the problem solving and not to the resource optimization. The code can be written in a cleaner way, with higher programming languages. Such a platform is easier to maintain and upgrade later. Electronic elements nowadays are developed very quickly and one moment sophisticated and expensive platform very quickly gets a successor, which is even more powerful. The price of the older platform then decreases and when the device is in production it is not so expensive anymore.

3. Design considerations

It is a good idea to take a lot of time at the beginning of the project to split the system into smaller logically separated units. The most important part is to define good boundaries between them. These are the so-called APIs (application programming interface). They should be as simple as possible, but must,implement all the needed functionality. The policy of how to use the APIs must be as simple as possible to avoid confusion and misunderstanding between developers. The most important API is the one that delimits the platform independent from the platform dependent code as shown in Figure 1. The simpler it is, the easier the porting to a new system will be if this is needed in the future. Additional well-defined APIs are used to delimit modules that will be assigned to different developers.

A good directory scheme of source code must also be designed, which fits well to the structure of developers' modules. In ideal case each module can be assigned a special directory, which will be maintained by one developer only. This is rarely a case and to avoid confusion and backup problems use of a version tracking system (CVS, Microsoft SourceSafe) is needed.

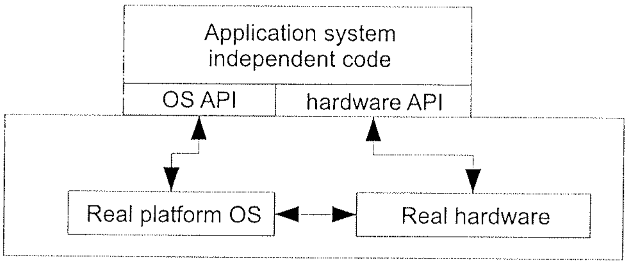


Figure 1: API used as a boundary between the platform independent and the platform dependent code.

A common programming 'technique' is use of 'copy and paste'. While this gives good enough results very quickly, it can be a source of hard to understand errors later. The reason is that when some part of such a code must be corrected, one usually forgets to make a correction on all the places where this code was also used. It is better to use functions performing the task (with parameters if the task is not exactly the same everywhere), macros, which will be expanded, or templates. Macros are deprecated because they produce name spacing problems and are hard to debug (breakpoints can usually not be set inside macros).

The higher the level of the programming language, the easier the programming is. The reason is that on the low level, the programmer must also focus on the correct use of code. The result is the distraction from the main problem, which the developers try to solve.

Humans can easily understand complicated data structures, especially when hierarchically organized, on the other hand it is very hard to trace all but the simplest program flows. Object oriented programming languages are specifically designed to make programming easier by focusing the programmers attention on data structures instead of algorithms (a case in classical programming).

Integrated debugging environments (IDE) with powerful graphical user interface are ideal for making an application very fast (rapid prototyping), but are less suitable when it comes to maintaining the code, reusing the code, automating the process of compilation, source saving, etc. Typical such programs make a lot of auto-generated code, which usually resides in predefined directories. Their configurations are typically in binary form, so they are less manageable then text based configurations. The only access possible is by mouse (sometimes it is very frustrating if a lot of clicking is needed to make a lot of identical changes, which could have been done with a simple 'find and replace'). Classical tools such as make, command line compilers and powerful script shells offer much greater flexibility but take a lot of time to learn how to use and setup. This time is very often saved later. They are very portable, flexible and easily automatically generated.

Data types used are very often a big problem when it comes to code portability. Apart from big/little endian compatibility (not so hard to manage properly) the most problems are caused by the difference of storage length and precision for simple data types (in C programming language), especially integer and sometimes also floating point data types. The same code that may work perfectly on one platform would fail - during runtime usually - on the other. There exist recommendations about this issue but care must always be taken to prevent problems. Using only specially defined (typedef) types helps a lot (using types directly from headers of different operating systems often introduces more confusion than it solves), but the type promotion rules in C always make problems.

The recommendations about the coding style and other rules may differ very much from person to person and cannot be generalized. Even related project teams do not always agree (/5/ and /6/). It is a good idea to read a lot of them so every valuable piece of information is taken into account. Very often some largely accepted rules just are not so efficient as generally thought, for example, extensive use of comments usually just puts additional load on developers, with no real benefit.

Multiple checking of the data validity takes additional effort to implement but may help reveal some errors. On both sides of an API a different range of data may be valid. For example, a key pressing is detected by the hardware and processed by the driver, which always implements some sort of glitch removal. At this stage typically only very quick (some microseconds) changes are considered as a glitch. This may or may not be suitable for the application, it depends on what is the behavior required by the customer. It therefore makes sense that the application implements its own data check algorithm (longer glitch detection in this case). Making so ensures the driver for the keyboard can remain the same regardless of the application requirements and is therefore useful also for future applications. However, the excessive use of double-checking leads to bad efficiency.

It is considered a bad programming technique to:

- Use global variables; instead, all data should be put locally on stack. This is slower and needs more memory but is much less error-prone. Even worse is reusing global variables for more than one purpose in order to save memory.
- Write code in an optimized way; it is better to concentrate on the clarity/readability of the code. That is especially true today when good compilers make much better optimization than any programmer could, and larger or slower final executable is not a problem either.
- Not to use operating systems, multi-threading, etc., except for extremely simple applications.

To make the code compile on many platforms two basic techniques are possible:

- Using the same modules with compilation switches.
- Using a common platform independent modules and separate modules for each platform that implements the dependent code.

The first approach is better only if the number of differences is very small (for example, using sockets under Unix or Microsoft Windows). Usually the second approach results in a more clear and easy to understand code. If the first approach is used, it is better to code the differences as macros or templates in a separate header in order to make the main code more readable.

The software documentation is often a problem. It is sometimes updated only few times during the development, for example, when explicitly required by the customer. It is not kept up-to-date when the changes in the software happen. The reason is almost always the time pressure. When there are not enough resources available to make the complete documentation, the priorities must be set up. It is usually sufficient to have a coarse description of the system architecture and algorithms used, which does not change much during the development. The details in the code should be commented in a way that the author, or any other normally skilled programmer, would understand them at any time in the future. Excessive use of comments is not a solution, doing so can make the code harder to understand, it is also very likely that the comments do not follow the code changes, which renders them misleading and wrong.

The APIs, on the other hand, deserve extremely detailed documentation, which consists not only of software interfaces (function prototypes, macros, enumerations, etc.), but also of detailed description of what conditions have to be fulfilled for the code on both sides of the interface in order to function properly (for example, thread safety policy, speed and processing capabilities limits). These conditions must be kept in mind for all the developers: those that implement the functionality should know what they must implement and what they need not to (but are allowed to if they can); those that use the functionality should be aware of API usage limitations and use only the documented features. It is very risky to count on the knowledge about the other parts of the system and to use non-documented features because they may change at any time. Any small change in APIs must be documented immediately.

4. Device simulation

Almost no development is possible without first making the product prototype. The same also applies for the embedded system software development. A powerful way to do this is simulating the behavior of the device and its environment.

Very often there exist simulators for target CPU and peripherals, which allow testing of native executables. Using one of them makes the debugging much easier. But the first level of simulation can be done in an environment used more generally by the developers.

The basic idea is to use a good compiler and debugger on a strong personal computer or workstation, to produce the first working prototype of the new application as a simulation. The application code (the platform independent code) is shared between this simulation and the real device that will be developed later in the process with a new platform as shown in Figure 2.

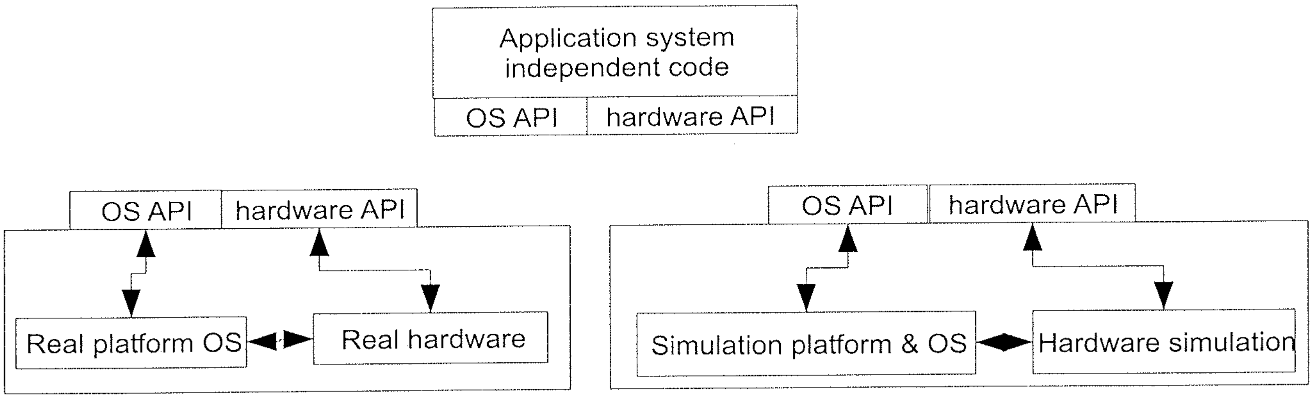


Figure 2: Replacement of the platform dependent code; the real platform and the simulated platform are interchangeable.

It must be noted that according to the general recommendations the prototype code should not be used in the final product, instead the code should be rewritten from scratch. Due to usual time pressures this is only seldom feasible.

This technique has one strong drawback - it is necessary to make an emulation of every part of the platform (operating system - OS, hardware, etc.) that will be present in the final product and set up one additional parallel project environment, which takes some time and efforts at the start of the project development. However, the simulation system usually turns out to be very simple; for example, in the real world sensing the state of a simple switch requires using an I/O port and writing the driver to handle this information, on the simulation that may be a simple button that is incorporated in a matter of seconds. LCD driver in a real world is complicated, but drawing bitmaps in any window environment is much easier.

Virtually all the other outcomes of this technique have a positive influence on the development, mainly because of the so-called front-loading effect /3/.

The main application development may be started long before the actual platform is available, tool packages set up and all the necessary drivers for the product are completed.

API between the system dependent code and the application, as well as any other APIs, can be evaluated and optimized. The API definition is more likely to be correct and complete if tested on many platforms. Any weak points or exceptions show up sooner and can be documented more reliably. This operation also forces the project manager and the programmers to think about how the code should be organized and split between the platform dependent and the platform independent part. The code organized this way is much easier to split between the developers and to port to a new hardware or OS platform if needed, since only the dependent part of the code has to be rewritten. This may happen sooner as predicted, because newer, better and more suitable platform elements emerge on the market very rapidly.

Debugging on the real platform is usually hard to do or is poorly supported. The code has to be loaded to the target, run and then connected with the host application. The ease of the debugging process depends on the debugging tool maturity, communication channel bandwidth that is established between the target and the host, etc. Available target simulators are platform oriented, which means a new simulator is usually needed if a new processor is chosen. In the case the producer remains the same, the existing simulator may still be useful, but changing to another producer usually means setting up an entirely different environment, which may not even have the same needed functionality.

Debugging the simulation on the personal computer or workstation is easy because standard, powerful and well-known tools are used. It is much faster, simpler and allows operations not possible on the real platform. Practically only the platform dependent code must be debugged directly on the target.

Device simulation can be connected to environment simulator, a program, which simulates the system where the device will be operating, to check if the behavior of the application is correct.

The simulation platform dependent code can be reused, usually with slight changes and enhancements, for the projects in the future.

Compiling the platform independent code on more than one hardware or OS platform and with different compilers can reveal warnings and errors that may only be discovered later during run-time, when they are much harder to track down. This way the code truly becomes independent.

It is very common that the development is started with insufficient analysis of the problem and customer needs, and continued with too weak emphasis on consistent software design basement. Instead, the coding starts very soon, usually because of tight schedule. This is hard to avoid. The development for parallel platforms forces this coding to be much more consistent and is thus less likely that corrections in the future will be needed.

The developers must resist the temptation to quicken the coding process by putting to the same module the parts of the code that are different in nature. This largely extends the complexity of such module and makes it much harder to understand, document and especially maintain. Generally, unless 'copy and paste' is used, porting the code to many platforms quickly discloses such coding style and forces the programmers to organize their code properly.

5. Environment simulation

Any electronic device typically operates in three basic stages:

- Capturing of the input data from the environment.
- Processing of the data.
- Making actions, returning the information to the environment.

Usually, the most complex part of the device software is the data processing. On the other side very often only small amount of the data is captured from the environment. That makes the simulation of such an environment extremely simple and easy to implement.

Example:

Electricity power metering device can capture the following input data:

- Voltage and current samples.
- Control inputs.

- Receiving characters over a serial communication.
- Detection of pressed keys on the console.

The information that is transferred to the output is:

- Data for the LCD display.
- Sending characters over a serial communication.
- Control outputs.

In this case input and output interface are very simple, compared to what this power meter device has to process:

- Calibrating the input data and calculating many different values: powers (active, reactive, apparent), voltages and currents (effective, maximum, minimum, harmonic analysis), frequency, etc.
- Tariffing and other processing of the data.
- Responding to input requests over communication and keys: parsing input data, output data formatting.
- Real time clock, security, checking data validity, etc.

For this system it is very simple to make the simulation of the environment. It consists of a simulating of the input signal samples, key presses and the communication channel, and on the output, of the LCD simulation and communication channel responses.

It turns out that such simplicity is very often a case. There may be some situations when simulating of the environment is hard to implement. For example, simulating the sensor data to the walking robot would be a complex task because of the strong feedback of robot's movements to the input sensors; this feedback is not trivial to simulate. But even in such cases the early simulation saves development resources /3/. First simulation may be low-fidelity and enhanced later, if needed. It can be made programmable and may support automated test procedures. This is useful during development, to test the new functionality, as well as for the device maintenance, when the device program is being changed because of fixing errors and sometimes adding new features. Namely, changing the program of the device may cause side effects that are not anticipated, and a complete system has to be tested as a consequence to ensure quality.

Besides testing of the device simulation, the environment simulator can be made to support the testing of the device on the real platform as well. The same test procedure is used; only the communication channel between the application and the environment simulation must be replaced. Pure simulation may run in the same executable as the environment simulation, communicating directly through the function calls. Real platform may connect to the environment simulator via one of available hardware communications (serial, Ethernet).

The general principle of how to incorporate the environment simulation to the system, is shown in Figure 3. The remote channel support represents the used communication means for connection to the environment simulation. In the Figure 3 the two hardware APIs implement the same functionality, which is identical to the general hardware API in the Figures 1 and 2. This way, the expanded application can directly replace the original application.

The switch can be realized as an object which all of the used information paths use to make the registration to. These information paths do not necessarily support all of the functionality, so they register itself each one with its own set of capabilities. Such implementation enables:

- Redirection of the output data from the application to all the data paths that support that data.
- Gathering input data from the data paths that can provide that type of data and deciding which input data path is in control at any given moment.

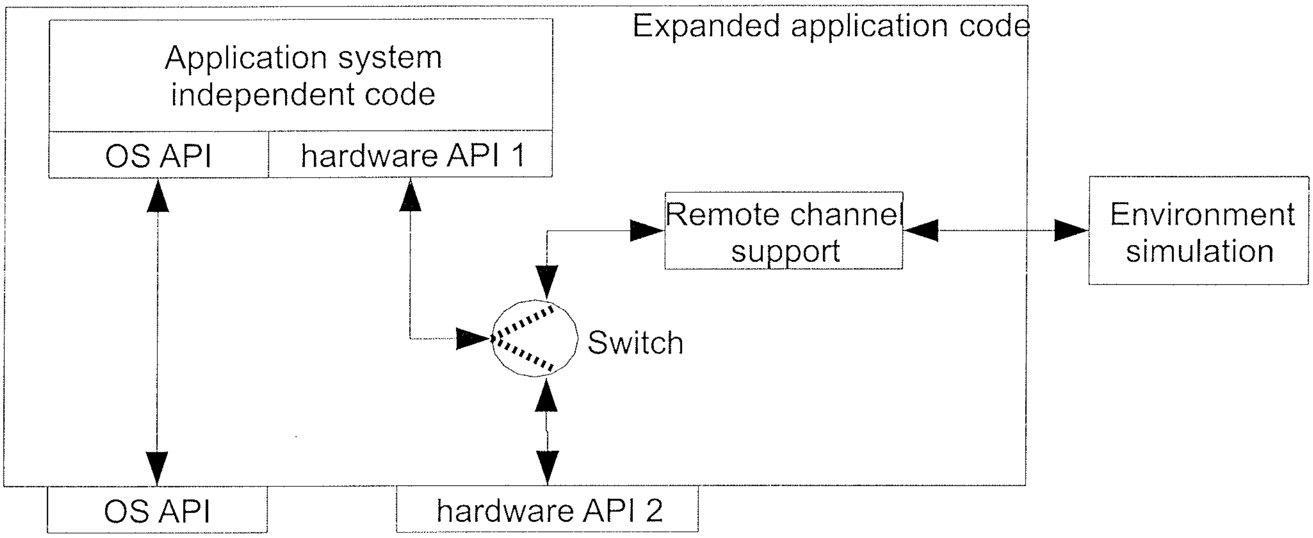


Figure 3: Insertion of a switch to support redirection of the data flow between the platform implementation and the remote environment simulation.

In Figure 3 only two data paths are shown, but this is not a limitation. For example, another path could be added to test a particular piece of hardware through the simulation environment.

Similar to the device simulation, using the environment simulation technique forces the developers to consider the application functionality even more in details. Some new ideas about how the input data could behave may be gathered. The whole system can be very successfully used as a demo that is presented to the customer. The customers very often do not have a good notion of how exactly the final system should behave and this is a good opportunity to compare wishes and reality. Demo can be presented in the earlier stages of the development when the changes are more easily implemented and are thus less expensive.

6. Conclusion

A situation in the software development domain was briefly described, stating that it is still far from reaching the point where high quality software is developed and delivered on time with all the promised functionality.

The simulation method of the application and the environment is particularly useful when developing the software for the embedded systems, especially where the input and output interface to the real world are not too complex, which is usually a case. These simulations can serve as a design aid during development and as a testing tool later.

All of these methods are not necessarily useful for every programmer, but they may be taken into account if they are found to fit into existing concepts of the development group. At least they may serve as an idea that would help produce newer, even better development methods. They are not meant to answer to the hard question how to consistently produce a good and reliably software on time, instead they provide some improvement in the case, which is very usual, when not enough development resources and time is available for the project.

7. References

/1/ F. P. Brooks, *The Mythical Man-month: Essays on Software Engineering*, Addison Wesley, MA, 1995, 0201835959

/2/ W. A. Sheremata, "Finding and Solving Problems in Software New Product Development", *Journal of Product Innovation Management*, 19 (2), 2002, 144-158.

/3/ S. Thomke, T. Fujimoto, "Shortening Product Development Time through 'Front-Loading' Problem-Solving", *CIRJE, Faculty of Economics, University of Tokyo*, CIRJE-F-11, http://ideas.repec.org/p/tky/fseres/98cf11.html.

/4/ P. G. Smith, "From Experience: Reaping Benefit from Speed to Market", *Journal of Product Innovation Management*, 16, 1999, 222-230.

/5/ http://www.purists.org/linux/: Linux Kernel Coding Style.

/6/ http://www.gnu.org/prep/standards.html: GNU Coding Standards

Stanislav Gruden
Iskraemeco d.d.
Savska loka 4, SI-4000 Kranj
E-mail: stanislav.gruden@iskraemeco.si

Prispelo (Arrived): 15.09.2003 Sprejeto (Accepted): 03.10.2003

HIGH-LEVEL SYNTHESIS BASED UPON DEPENDENCE GRAPH FOR MULTI-FPGA

Mohamed Akil

Laboratoire A2SI, Groupe ESIEE, Cité Descartes, Noisy Le Grand cedex, France

INVITED PAPER
MIDEM 2003 CONFERENCE
01.10.2003 - 03.10.2003, Grad Ptuj

Abstract: The increasing complexity of signal, image and control processing algorithms in real-time embedded applications requires efficient system-level design methodology to help the designer to solve the specification, validation and synthesis problems. Indeed the real-time and embedded constraints may be so strong that the available high performant processors are not so enough. That leads to use, in complement of processor, the specific component like ASIC or FPGA. Several projects have developed high-level design flow that translates high-level algorithm specification to an efficient implementation for mapping onto multi-component architecture. In this paper, we present: 1. a unified model for hardware/software codesign, based on the AAA methodology (Algorithm-Architecture Adequation). In order to exhibit the potential parallelism of algorithm to be implemented, the AAA methodology is based on conditioned (conditional execution of computations) factorized (loop) data dependence graph, 2.Some simple rules that allow synthesizing both the data path and the control path of a circuit corresponding to an algorithm specified as a Conditioned and Factorized Data Dependence Graph (CFDDG), 3. the optimized implementation of CFDDG algorithm onto FPGA circuit and Multi-FPGA (partionning), by using simulated annealing approach, 4. the resources and time delay estimation method. This method allows us to have a performance analysis for the implementation. The obtained results: resource (gates, IO) and latency estimation are used by the optimization step to decide which implementation respects the constraints (real-time implementation which minimises the resource utilisation), 5. the results of the implementation of the matrix-vector product algorithm onto a Xilinx Multi FPGA and the software tool SynDEx which implements the AAA methodology.

Visokonivojska sinteza FPGA vezij na osnovi odvisnih grafov

Izvleček: Naraščajoča zapletenost algoritmov za obdelavo signalov, slike in opravljanje nadzora v vgrajenih sistemih v realnem času zahteva učinkovite metodologije načrtovanja na nivoju sistema z namenom pomagati načrtovalcu reševati probleme pri specifikaciji, validaciji in sintezi sistema. V resnici se lahko zgodi, da so omenjene zahteve tako zahtevne, da omejijo uporabo obstoječih zmogljivih procesorjev. To navede na uporabo dodatnih specifičnih komponent, kot so ASIC vezja ali FPGA vezja. Pri nekaj projektih se je že zgodilo, da smo uspeli zahteve algoritmov na visokem nivoju prevesti na implementacijo arhitekture zasnovane na sistemu z večimi komponentami. V tem prispevku predstavimo : 1. poenoten model za sočasno načrtovanje programske in strojne opreme zasnovane na metodologiji AAA (Algorithm-Architecture-Adequation), ki je zasnovana na pogojnem podatkovno odvisnem grafu s čimer lahko izkoristimo potencialno paralelno izvajanje algoritma. 2. nekaj osnovnih pravil, ki omogočajo sintezo podatkovnih in kontrolnih poti za vezje, ki odgovarja algoritmu definiranemu kot CFDDG graf. 3. optimizirano implementacijo CFDDG algoritma z enim ali več FPGA vezji z uporabo metode simuliranega ohlajanja. 4. sredstva in metodo za oceno časovnih zakasnitev. Ta metoda omogoča analizo delovanja za določeno implementacijo. Dobljeni rezultat : sredstva (vrata, IO) in oceno latentnosti uporabimo pri koraku optimizacije za odločanje, katera implementacija spoštuje omejitve (implementacija v realnem času, ki minimizira uporabo sredstev). 5. rezultate implementacije algoritma matričnega produkta na Xilinx Multi FPGA vezje in programsko orodje SynDex s katerim je izvedena AAA metodologija.

1. Introduction

As the size and complexity of high performance signal, image and control processing algorithms is increasing continuously, the implementations cost of such algorithms is becoming an important factor. This paper addresses this issue and presents an efficient rapid prototyping methodology to implement such complex algorithms using reconfigurable hardware. The proposed methodology is based on an unified model of conditioned factorized data dependence graphs where both data and control flow are represented as well to specify the application algorithm, than to deduce the possible implementations onto reconfigurable hardware, in terms of graphs transformations. This work is part of the AAA methodology and has been implemented in SynDEx (CAD software tool that support AAA, a system level CAD software tool.

To fulfill the ever increasing requirements of embedded real-time applications, system designers usually require mixed implementation that blends different types of programmable components (RISC or CISC processors, DSP,..)corresponding to software implementation, with specific non-programmable components (ASIC, FPGA,...) corresponding to hardware implementation.

This makes the implementation task a complicated and challenging problem, which implies a strong need for sophisticated CAD tools based on efficient system-level design methodologies to cope with these difficulties and so to simplify the implementation task from the specification to the final prototype.

In this field, several system-level design methodologies and their associated tools have been suggested during the last years. SPADE /1/ methodology enables modelling and exploration of signal heterogeneous processing systems. The result is the definition of a heterogeneous architecture able to execute these applications with respect real-time constraints. SPARK /2/ is high-level synthesis framework that provides a number of code transformations techniques. The back-end of the SPARK system generates synthesizable RTL VHDL (control synthesis is a finite sate machine controller). GRAPE-II /3/ is a system-level development environment for specifying, compiling, debugging, simulating and emulating digital-signal processing applications on heterogeneous target platforms consisting of DSPs and FPGAs. After specification, resources requirement, mapping architecture, the last phase generates C or VHDL code for each of the processing devices. POLIS /4/ system implements a HW/SW codesign using the CFSM (the Codesign Finite State Machine formal model). A complete codesign environment, based on POLIS system, which combines automatic partitioning and reuse of a module database is presented in polis. The SPARCS design system /5/ is an integrated design environment for automatically partitioning and synthesizing behavioural specifications (in the form of task graphs) on multi-FPGA architecture. The SPARCS contains a temporal partitioning tool to temporally divide and schedule the tasks on the architecture and high-level synthesis tool to synthesize register_transfer level designs for each set of tasks.

Each of the above tools has its own features (for example several models can be used for application and architecture specification) and innovative aspects but none of them support the entire implementation process onto mixed architecture using an unified model as well to specify the application algorithm, as to deduce the possible implementation onto multicomponent architecture.

To achieve this goal, we have developed, in the one hand, the AAA (Algorithm-Architecture Adequation) rapid prototyping methodology /6/ which helps the real-time application designer to obtain rapidly an efficient implementation of his application algorithm onto his heterogeneous multiprocessor architecture and to generate automatically the corresponding distributed executive /7/. This methodology uses an unified model of graphs as well to modelize the application algorithm, the available architecture as to deduce the implementation which is formalized in terms of transformations applied on the previous graphs. In the other hand we aim to extend our AAA methodology to the hardware implementation onto specific integrated circuits in order to finally provide a methodology allowing to automate the implementation of complex application onto multicomponent architecture using an unified approach.

This paper presents the design methodology based upon graph transformation from algorithm specification to hardware implementation. This methodology automates the hardware implementation of an application algorithm specified as a Conditioned Factorized Data Dependence graph in the case of reconfigurable integrated circuits (FPGA). This methodology is illustrated through all the sections with a conditioned matrix-vector product case study that involve a moderately complex control flow involving both conditioning and loops. We first present the conditioned factorized data dependence graph model proposed to specify the application algorithm in section 2. In section 3 we present the implementation model describing the result obtained by applying a set of rules that allows to automate the synthesis of data and control paths from the algorithm specification. Following that, the principles of optimization by defactorization are described in section 4. In this section we present the using of the simulated annealing technique to obtain an optimized implementation on mono and multi circuit architecture. The proposed algorithms guided by the cost functions find the best solution that respects the real time constraint while minimizing the resources consumption.

2. AAA methodology: Algorithm model

According to the AAA methodology, the algorithm model is an extension of the directed data dependence graph, where each node models an operation (more or less complex, e.g. an addition or a filter), and each oriented hyperedge models a data dependence, where the data produced as output of a node is used as input of an other node or several other nodes (data diffusion). The set of data dependences defines a partial order relation on the execution of the operations, which may be interpreted as a "potential parallelism".

This extended data dependence graph, called Conditioned Factorized Data Dependence Graph (CFDDG) allows to specify loops through factorization nodes, and conditioned operations (operation executed, or not, depending on its conditioning input) through conditioning edge. In this CFDDG graph, each oriented dependence edge is either a data dependence or a conditioning dependence, and each node is either a computation operation, an input-output operation, a factorization operation or a selection operation.

This algorithm graph may be specified directly by the user using the graphical or textual interface of the SynDEx software / 7/ or it may be generated by the compiler from high level specification languages, such as the synchronous languages, which perform formal verifications in terms of events ordering in order to reject specifications including deadlocks /8/.

2.1 Conditioned Factorized Data Dependence Graph

Typically an algorithm specification based on data dependence contains regular parts (repetitive subgraph) and nonregular parts. As described in /9/, these spatial repetitions of operation patterns (identical operations that operate on different data) are usually reduced by a factorization process to reduce the size of the specification and to highlight its regular parts. Graph factorization consists in replacing a repeated pattern, i.e. a subgraph, by only one instance of the pattern, and in marking each edge crossing the pattern frontier with a special "factorization" node, and the factorization frontier itself by a doted line crossing these nodes. The type of factorization nodes depends on the way the data are managed when crossing a factorization frontier: 1. A Fork ***F*** node factorizes array partition in as many subarrays as repetitions of the pattern. 2. A Join ***J*** node factorizes array composition from results of each repetition of the pattern. 3. A Diffusion ***D*** node factorizes diffusion of a data to all repetitions of the pattern. 4. An Iterate ***I*** node factorizes inter-pattern data dependence between iterations of the pattern.

Moreover, the user may want to specify that some operations will be executed depending on some condition. In our CFDDG model, we provide a conditioning process such that the execution of operations of the algorithm graph may be conditioned by a conditioning dependence, which is represented on the algorithm graph by a dashed edge. In this case, the conditioned operation is executed only if its inputs data are present and its condition of activation is satisfied. In order to indicate the end of the conditioned sub-graph in the algorithm graph that corresponds to the 'EndIf' of the typical control primitive IF-THEN-ELSE, we need a specific node 'select'. It allows to select among the data it receive the one that will be sent to its output. The input data of a select node correspond to the data produced by the conditioned operations with their condition of activation satisfied. As the parallel execution of these conditioned operations, that are not necessarily exclusive, can lead to simultaneous presence of several input data at the select node, we introduced priorities between its data which will be specified in an explicit way with labels on the input edges (p_1, p_2, ...,p_n). The input data having the highest priority p_i will be selected and sent to its output.

2.2 Specification of Conditioned Matrix-Vector by using Conditioned Factorized Data Dependence Graph

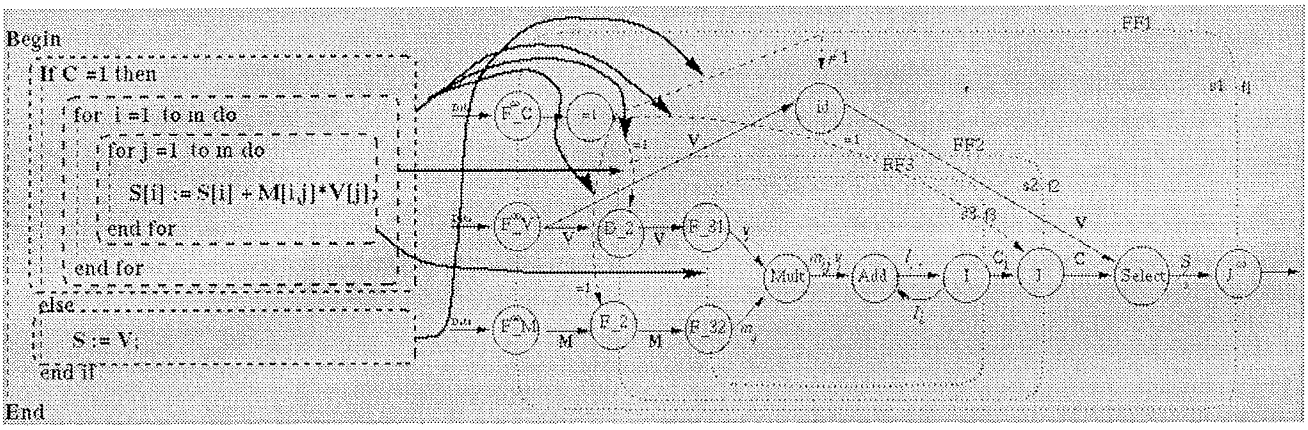
Figure 1 represents use a Conditioned Matrix-Vector Product example (C-MVP) specifying by CFDDG model: depending on the value of the input data C, this algorithm will compute either the product of the matrix $M \in R^m x R^n$ by the vector $V \in R^n$ and will return the resulting vector or will directly return the input vector V. The computation of the product of the matrix M (composed of m vectors M_i: $M = (M_i)_{1 \leq i \leq m}$ by the vector V can be decomposed into m scalar products $PS = (M_iV)_{1 \leq i \leq m}$ (loop for i) each PS can then be decomposed into a sum of n products $M_iV = M_{i1}V_1 + ... + M_{in}V_n$ (loop for j).

This decomposition process generates repetitions of operations patterns; that we often prefer to specify in a factorized form as described in Fig.1. Therefore, the final conditioned Factorized Data Dependence Graph (CFDDG) will include the two imbricated frontiers FF2 and FF3 corresponding to the two imbricated for loops, in addition to the factorization frontier FF1 which correspond to the factorization of the infinitely repeated pattern of the graph since we deal with reactive applications that interact infinitely with the physical environment.

3. AAA methodology: Implementation model

Implementing applications onto specific integrated circuits requires system designers to generate the data path responsible for the core of the computation as well as the control path to provide the appropriate control signals for the computations. The resulting RTL design containing both data and control paths is then characterized in order to estimate time and area performance. This allows the exploration of different hardware implementations, seeking for an ideal compromise between the area and the response time of the circuit.

Then, we propose a seamless flow based on graph transformation to transform the algorithm graph into an implementation graph containing both data-path graph and control-path graph. As will see, data-path transformations are quite simple, but control-path transformations are not trivial and require to build first a neighborhood graph.



```
Begin
  If C =1 then
    for i =1 to m do
      for j =1 to m do
        S[i] := S[i] + M[i,j]*V[j];
      end for
    end for
  else
    S := V;
  end if
End
```

Figure 1: Conditioned and Factorized Data Dependence Graph of C-MVP

3.1 Neighborhood graph

Every factorization frontier may be a consumer (located downstream) or/and a producer (located upstream) relatively to another frontier according to the data dependences relating them. Two frontiers are neighbors if there is at least one relation of direct dependence that does not cross a third frontier. Based on these neighborhood relations, we build a neighborhood graph. The nodes of such graph represent the factorization frontiers and the oriented edges represent the data flow between factorization frontiers.

The edge orientation describes the consumption/production relation: an edge starts at a producer and ends at a consumer. As producing/consuming frontiers may be them selves conditioned (e.g. FF2 on Fig.1), data production/ consumption between frontiers are consequently conditioned. To take into account such conditioned data flow, we will represent conditioned consumption/production by dashed edges.

In the case of a sequential implementation of factorization nodes, every factorization frontier, called FF, separates two regions, the first one called "fast" (f), being repeated relatively to the second one, called "slow" (s). These slow and fast sides of a frontier are due to the difference of the data transfer rate on each side of the frontier. Every node of the neighborhood graph is then subdivided in four parts (slow-downstream, fast-upstream, fast-downstream and slow-upstream) /10/. The neighborhood graph is deduced automatically from the CFDDG and it is used during the implementation in order to establish the control relationships between frontiers leading to a part of the control-path.

The neighborhood graph built from the CFDDG specifying the C-MVP algorithm (Fig.1) comprises three nodes corresponding to the three factorization frontiers FF1, FF2, FF3. The factorization frontier FF1 is infinite, it does not have neighbors on its "slow" side which corresponds to the physical environment. FF1 is, either a producer compared to the conditioned frontier FF2 or a producer to itself and a consumer compared either to itself or to the conditioned frontier FF2. FF2 is also a producer and a consumer compared to FF3. FF3 is a producer and a consumer, compared to itself through the arithmetic operations mul and add.

3.2 Data-path graph generation

The hardware implementation of the Conditioned Factorized Data Dependence Graph consists in providing a matching operator for every node, and a matching connection between operators for each data dependence edge relating the corresponding operations. The resulting graph of operators and their interconnections compose the data path of the circuit. This hardware translation process defines then a graph isomorphism between Conditioned Factorized Data Dependence Graph and the data path graph.

The matching operator node is a logic function in the case of a computation operation node, or it is composed of a multiplexer and/or registers in the case of a factorization node (i.e: ***F***, ***I*** and ***J*** nodes) or it is composed of a priority encoder and a multiplexer for the select node to encode priority and to select data.

3.3 Control-path graph generation

The control path corresponds to the logic functions that must be added to the datapath, in order to control the multiplexers and the transitions of the registers composing the operators. It is then obtained by data transfer synchronization between registers. However, two conditions must be satisfied to allow a register to change state: the new upstream data to the register must be stable, and all downstream consumers of the register must have finished the utilization of previous data. Moreover, if upstream data of a circuit comes from various producers with different propagation time, it is necessary to use a synchronized data transfer process. This synchronization is possible through the use of a request/acknowledge communication protocol. Consequently, the synchronization of the circuit implementing the whole algorithm is reduced to the synchronization of the request/acknowledge signals of the set of factorization operators.

These operators are gathered in factorization frontier and their data consumption and production are done in a synchronous way at the level of the frontier. We propose then a local control system where each factorization frontier will have its own control unit.

This delocalized control approach allows the CAD tools used for the synthesis to place the control units closer to the operators to control rather then a centralized control approach.

Control units and their interconnections: As mentioned above (section 3.1), each factorization frontier has upstream and downstream relations on both sides, "slow" and "fast". The relations between upstream/downstream and request/ acknowledge signals on both sides of a frontier are implemented by the "control unit" of the factorization frontier. This control unit contains a counter C with d states (corresponding to the d factorized repetitions) and an additional logic function in order to generate, in the one hand the communication protocol between frontiers (the slow/fast, request/ acknowledge signals at the upstream and downstream sides), and in the other hand the counter value (cpt) and the enable signal (en), that control the frontier operators.

Thus, the control path will mainly be composed of the set of control units associated to the corresponding frontiers nodes of the neighborhood graph. These control units are then inter-connected in a systematic way as follows: for each oriented dependence edge, we generate a request signal transmitted between the corresponding control units. And for each generated request signal, the associated acknowledge signal is transmitted, in the opposite direc-

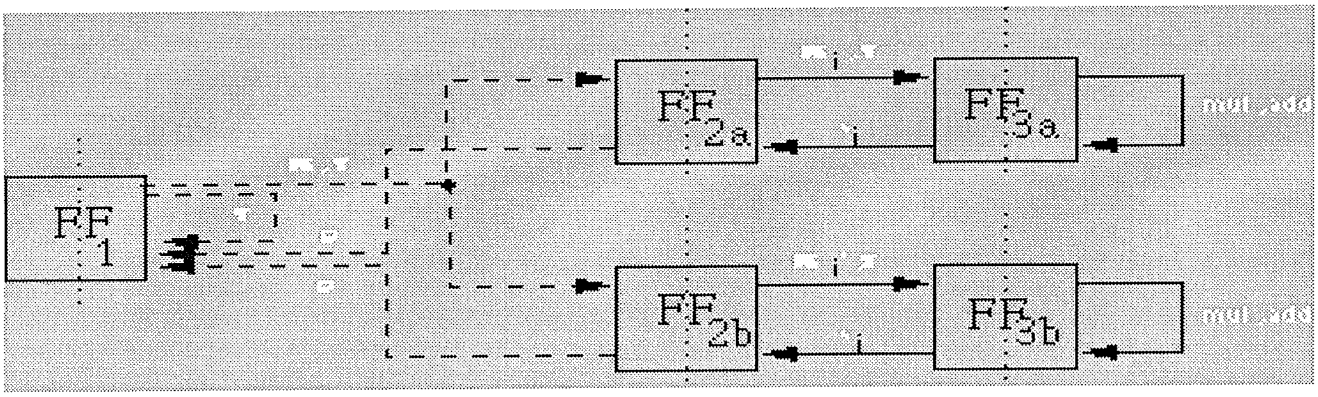


Figure 2: Neighborhood graph of defactorized C-MVP (see paragraph 2.2)

tion. When several signals occur at the same input of a control unit, the conjunction of these signals is performed by a logical AND gate. Note that, the generated request signals associated to conditioned dependences must first be send to a multiplexer controlled by a priority encoder which will send in turn the request signal with the highest priority to its output /11/.

4. AAA methodology: Implementation optimization

4.1 Optimization principle based upon defactorization process

If the implementation of the factorized specification onto an application specific integrated circuit or an FPGA does not meet the real time constraints, we need to defactorize the implementation graph corresponding to the specification. The defactorization process is the reverse transformation of the factorization and therefore it does not change the operational semantic of the data dependence graph. The goal is to obtain a more parallel implementation in order to reduce the latency and improve the temporal performances in spite of increasing hardware resources. Thus the optimized implementation of a conditioned factorized algorithm graph onto the target architecture is formalized in terms of graph defactorization transformation.

Figure 3 represents a defactorized by a factor 2 of C-MVP graph. Defactorized solution allows to reduce the latency of the implementation, but increase the number of required hardware resources. FF2 is defactorized in two frontiers FF2a and FF2b, and FF3 is then duplicated in FF3a and FF3b.

The implementation space which must be explored in order to find the best solution is then composed of all the possible defactorizations of a factorized graph specifying the algorithm. For instance, for a given algorithm graph with n frontiers, we have at least 2^n defactorized implementations. Moreover, each frontier can be partially defactorized: a factorization frontier of *r* repetitions can be decomposed in *f* factorization frontiers of *r*/*f* repetitions. Consequently, for a given algorithm graph, there is a large, but finite, number of possible implementations which are more or less defactorized, and among which we need to select the most efficient one, i.e. which satisfies the real-time constraints (upper bound on latency), and which uses as less as possible the hardware resources, logic gates for ASIC and number of Configurable Logic Blocks CLB for FPGA. This optimization problem is known to be NP-hard, and its size is usually huge for realistic applications. This is why we use heuristic guided by a cost function, in order to compare the performances of different defactorizations of the specification. This heuristic allows us to explore only a small subset of all the possible defactorizations into the implementation space. The heuristic needs to define a cost function based on the critical path length metric of the implementation graph: it takes into account both the latency and the resources consumption of the implementation which are obtained by a preliminary step of characterization.

4.2 Architecture characterization: area and latency estimation

To estimate the total area we use the neighborhood graph to calculate the data path area and control path area. This total area used by the implementation is given by:

$$S_t(FF,FG)=\sum_{i=1}^{n} D_{path}(f_i)+\sum_{i=1}^{n} S_{Cpath}(f_i) \qquad (1)$$

Where f_i designs the frontier i.

The total area is calculated in term of FF (Flip Flop) and FG (Function Generator). One can use equation 2 to deduce the area in terms of CLB (Control Logic Block):

$$S_t(CLB)=\frac{\max(S_t(FF),S_t(FG)}{2} \qquad (2)$$

For the calculation of latency, one deduces from neighbourhood graph the various relations between the frontiers (frontiers in series, parallel, inclusive). These relations enable us to determine the number of cycles, thus the number of cycles multiplied by the time cycle gives the execution time of the algorithm.

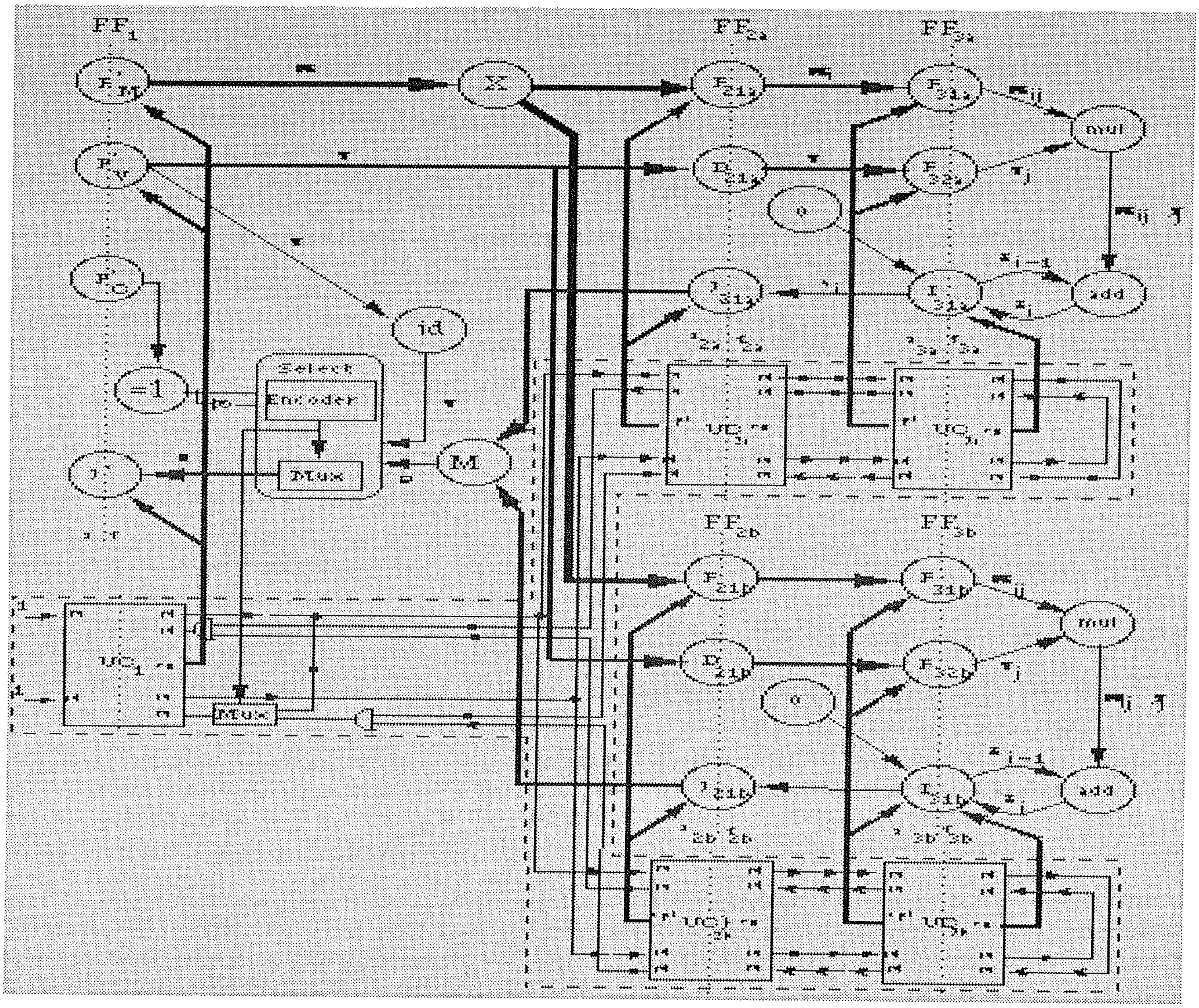


Figure 3: A defactorized implementation graph of C-MVP

4.3 Algorithm of the proposed Simulated annealing heuristic for mono-FPGA architecture

4.3.1 Simulated annealing principle

This is a technique /12/ for solving combinatorial optimisation problems such as minimising functions of many variables. The conventional ANNEALING algorithm operates by starting with an initial solution to the combinatorial optimization problem and improving the solution through a series of changes. This involves finding some configuration of parameters that minimises some objective function. It is based on a modification of iterative improvement, which involves starting with some existing suboptimal configuration and perturbing it in some small way. If this new configuration is better than the old one the new configuration is accepted and we start again. Unlike a greedy algorithm which only accepts system configuration resulting from better changes, annealing probabilistically accepts inferior changes.

4.3.2 Application the simulated annealing technique in the AAA methodology

The application of simulated annealing consists in finding the global minimum (i.e. the optimized implementation graph) of an objective function by avoiding its local minima (metastable states of the system). This function has two variables: resources used for the hardware implementation and execution time of the algorithm.

Let a vector containing a set of variables, each one of these variables indicates the state of each component of the system. In our case the system is represented by the CFGDD of the algorithm to be implemented, each frontier of this graph defines a component of the system and a variable X defines the defactorization factor. The defactorization process implies a change of the system. For example: if the V1 vector = {X1=3, X2=1, X=3,X4=2) changes into in V1 = {X1=3, X2=2, X3=3, X4=2), the new state of V1 means that the frontier which is defined by X2 was defactorized by 2. As we described in paragraph 4.1, this defactorization process generates an increasing/diminution of the execution time of the algorithm (i.e. latency of the circuit) and thus a diminution/increasing of the surface of the circuit. We defined a cost function F(X) for a given state X of the system. This function estimates the variations of the system for a given state X. It permits to choose the defactorization which satisfies the time constraint while minimizing the resources consumption (i.e. CLB in case FPGA):

$$F(X)=\begin{cases} S & \text{if } t \prec T \\ S+k(t-T) & \text{if } t \geq T \end{cases} \qquad (3)$$

Where:

- T indicates the time constraint,
- t is the execution time for a given defactorization,
- S is the surface of the hardware implementation for a given defactorization,
- k defines the penalty factor.

Based on the technique of simulated annealing, the algorithm that we propose starts with the calculation of the control parameter C_0, then one starts by gradually decreasing his value and for each one of these values one carries out a certain number of changes of states of the system (defactorization). With each reduction of the control parameter of control, one increases the number of changes of states of the system by a factor β.

In the algorithm below:

- *X_0* presents the initial solution of the system,
- *L_0* is the initial value of a number of changes of system state,
- *iter* defines the number of change of a control parameter,
- *L_k* is the modification number of system state for a given control parameter c_k,
- *X_i* is a given solution or given state vector of the system,
- *X_j* is state vector after a defactorization process, it is a solution close to X_i.
- *c_k* is indicates the value of a control parameter during the kth iteration, is the initial value of a control parameter,
- *F(X)* indicates the cost function for a system state X.

Algorithm: simulated annealing for mono circuit architecture

```
initialize ($c_0$,$L_0$,$X_0$ )
for iter =1 to num_iter do
    for n-modif to $L_k$ do
        $X_j$=V($X_i$)
        if (F($X_i$)≺F($X_j$)) then $X_i$=$X_j$
        else
            if(exp(-(F($X_j$)-F($X_i$))/c)≻Random.float(0,1)) then $X_i$=$X_j$
            endif
        endif
    endfor
    $C_{K+1}=C_K*\alpha$
    $L_{K+1}=L_K*\beta$
endfor
```

The simulated annealing algorithm returns the optimized implementation graph which satisfies real-time constraints and uses as less as possible the hardware resources. The algorithm begins by performing the initial value of X_0, L_0 and c_0 (c_0 is equal to 2*n, where n defines the number of defactorizable frontiers) and then decreases gradually a control parameter. For each control parameter, the state system is modified and for each control parameter reduction, the number of the system change is increased by β. One chooses β equal to the edges number of the CFD-DG. The algorithm must find and accepts the solution close to X_i if it is not possible another solution is accepted with certain probability.

4.4 Algorithm of the proposed Simulated annealing heuristic for Multi-FPGA architecture

Let the number of frontiers composing the graph algorithm CFGDD, each frontier border of this graph contains a set of edge and has a factor of factorization.

A X vector defines the different states of the frontier defactorization. Let FF2 frontier which is defined by X2 variable. If FF2 frontier has a factor defactorization equal to 3 then X2 contains three elements: X21, X22 and X23. Each of these elements corresponds to a defactorization state. For

$$F(X)=\begin{cases}
S_{tot}+I/O_{tot} & \text{if } t_{tot}<T_{contr} \text{ and } \forall S_i<S_{contr} \text{ and } \forall I/O_i<I/O_{contr} \\
S_{tot}+I/O_{tot}+k(t_{tot}-T_{contr}) & \text{if } t_{tot}\geq T_{contr} \text{ and } \forall S_i<S_{contr} \text{ and } \forall I/O_i<I/O_{contr} \\
S_{tot}+g.(I/O_i-I/O_{contr})+k(t_{tot}-T_{contr}) & \text{if } t_{tot}\geq T_{contr} \text{ and } \forall S_i<S_{contr} \text{ and } \exists I/O_i\geq I/O_{contr} \\
l(S_i-S_{contr})+I/O_{tot}+k(t_{tot}-T_{contr}) & \text{if } t_{tot}\geq T_{contr} \text{ and } \exists S_i\geq S_{contr} \text{ and } \forall I/O_i<I/O_{contr} \\
l(S_i-S_{contr})+g(I/O_i-I/O_{contr})+k(t_{tot}-T_{contr}) & \text{if } t_{tot}\geq T_{contr} \text{ and } \exists S_i\geq S_{contr} \text{ and } \exists I/O_i\geq I/O_{contr} \\
l(S_i-S_{contr})+g(I/O_i-I/O_{contr}) & \text{if } t_{tot}<T_{contr} \text{ and } \exists S_i\geq S_{contr} \text{ and } \exists I/O_i\geq I/O_{contr} \\
l(S_i-S_{contr})+I/O_{tot} & \text{if } t_{tot}<T_{contr} \text{ and } \exists S_i\geq S_{contr} \text{ and } \forall I/O_i<I/O_{contr} \\
S_{tot}+g(I/O_{tot}-I/O_{contr}) & \text{if } t_{tot}<T_{contr} \text{ and } \forall S_i<S_{contr} \text{ and } \exists I/O_i\geq I/O_{contr}
\end{cases} \qquad (4)$$

example, if one defactorises FF2 by 2 then X22 is equal to 2 and X21 and X23 are equal to 0. X22 contains a set of frontiers which their edges.

To each frontier, one associates in random way a given circuit of the architecture by applying the simulated annealing method. In order to estimate the partionning we defined a cost function:

Where :

- t_{tot} is the execution time of the algorithm after applying defactorization,
- S_{tot} is the area occupied by all the circuits after the defactorization process
- S_i defines the area used in the circuit i,
- k,l,g are the penalty coefficients (varying from 10 to 100) used if respectively the time, area and I/O constraint are not respected,
- T_{contr} is the time constraint,
- S_{contr} and I/O_{contr} define respectively the time and I/O constraints for every implementation,
- I/O_{tot} is the total number of the I/O used for all the circuits used by a given implementation,
- I/O_i is the I/O number for the circuit *i*.

The proposed algorithm is similar than the mono circuit case, we only change the cost function. This algorithm takes the neighbourhood graph as an input and returns the partionning graph. This partinoning graph corresponds to muli-circuit implementation which respects the real time constraints and uses as less possible the hardware resources (i.e. area and I/O).

We applied this algorithm to an example which describes a five Conditioned Product Matrix Vector calculation (the CFDD graph contains 60 edges). The hardware implementation uses two circuits: Xilinx XC4013E/X2. The parameters bellow defines the main characteristics of XC 4012 E/X2 circuit:

Device	Logic Cells	Max Logic Gates	Max RAM bits	Typical Gate Range	CLB Matrix	Total CLBs	Number of Flip - Flops	Max User I/O
XC4013E/X2	1,368	13,000	18,432	10,000 - 30,000	24x24	576	1,536	192

To compare the optimisation results, we applied the simulated annealing algorithm for 5 different time constraints: 1000, 1500, 2000, 2500 and 3000 ns. The table below presents the results for each time constraint. The hardware implementation needs two circuits: circuit 1 and circuit 2. For each circuit table 1 gives the CLB and I/O used at different time constraint.

5. Conclusion

We showed that from an application algorithm specified with a conditioned factorized data dependence graph it is possible to obtain a hardware implementation onto a reconfigurable integrated circuit following a set of graphs transformations, leading to a seamless design flow. These transformations allow to automatically generate the datapath and the control-path for designs with moderately complex control flow involving both conditioning and loops. The proposed delocalized control approach allows the CAD tools used for the synthesis to place the control units closer to the operators to control. We have presented an optimization heuristic based upon simulated annealing and we applied this technique for Conditioned and factorized Data Dependence Graph by using a defactorization process guided by cost function. We defined two cost functions for mono and multi Circuit architectures. We used these two algorithms to obtain automati-

TIME Constraint [ns]	1000	1500	2000	2500	3000
circuit 1 CLB used	60	433	298	202	306
Circuit 2 CLB used	458	114	210	317	172
circuit 1 I /O used	34	30	41	35	43
Circuit 2 I/O used	37	27	39	30	47
Total latency [ns]	397	1231	985	1280	1231
Total CLB	518	547	508	519	478
Total I/O	71	57	80	65	90

Table 1: number of circuit, CLB and I/O used at different time constraint.

cally the best solution at the given time constraint. The optimization heuristics will address both defactorization and partitioning issues. Moreover, this extension of the AAA methodology to the hardware implementation of algorithm onto integrated circuit, provides a global methodology in order to tackle complex hardware/software co-design problems involved by multicomponent architecture.

6. References

/1/ P. Lieverse, P. van detr Wolf, Ed Deprettere, K. Vissers, "A Methodology for architecture exploration of heterogeneous signal processing systems", Proc. 1999 IEEE Workshop on Signal Processing Systems.

/2/ S. Gupta, N. Dutt, R. Gupta, A. Nicolau, "SPARK, High-Level Synthesis Framework For Applying Parallelizing Compiler Transformations", 7th Intl.Conference on VLSI design, January 5-9, 2004, Mumbai, India.

/3/ R. lauwereins, M. Engels, M. Adé, J. Peperstraete, "Grape-II : A system-level Prototyping Environment For DSP applications", IEEE Computer, Vol. 28,No 2, pp. 35-43, Feb. 1995.

/4/ S. Edwards, L. Lavagno, E.A. Lee, A. Sangiovanni-Vincentelli, "Design of embedded systems: formal models, validation, and synthesis", Proc. of IEEE, v.85, n.3, March 1997.

/5/ M. Kaul , R. Venuri, "Optimal Temporal Partitioning and Synthesis for Reconfigurable Architectures", Design, Automation and Test in Europe, February 1998.

/6/ T. Grandpierre, C. Lavarenne, Y. Sorel, "Optimized rapid prototyping for real-time embedded heterogeneous multiprocessors", CODES'99 7th Intl. Workshop on Hardware/Software Co-Design, Rome, May 1999.

/7/ T. Grandpierre, Y. Sorel, "From algorithm and architecture specifications to automatic generation of distributed real-time executives: a seamless flow of graphs transformations", MEMOCODE03, Intl. Conference on Formal Methods and Models for Codesign, Mont Saint-Michel, France, June 2003.

/8/ N. Halbwachs, "Synchronous programing of reactive systems", Kluwer Academic Publishers}, Dordrecht Boston, 1993.

/9/ L.Kaouane, M. Akil, Y. Sorel, T. Grandpierre, "From algorithm graph specification to automatic synthesis of FPGA circuit: a seamless flow of graphs transformations", FPL'03, Intl.Confrence on Field Programmable Logic and Applications}, Lisbon, Portugal, September 1-3, 2003.

/10/ L.Kaouane, M. Akil, Y. Sorel, T. Grandpierre, "A methodology to implement real-time applications on reconfigurable circuits", ERSA'03, Intl. Conference on Engineering of Reconfigurable Systems and Algorithms}, Las vegas, USA, June 2003.

/11/ R. Vodisek, M. Akil, S.Gailhard, A.Zemva, "Automatic Generation of VHDL code for SynDEx v6 software", Electro technical and Computer Science conference}, Portoroz, Slovenia, September 2001.

/12/ S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, „Optimization by Simulated Annealing", Science, 220(4598): 671-680, May 1983.

Mohamed Akil
Laboratoire A2SI, Groupe ESIEE
Cité Descartes, 2 Bld Blaise Pascal - BP 99
93162 Noisy Le Grand cedex, France
akilm@esiee.fr

Prispelo (Arrived): 15.09.2003 *Sprejeto (Accepted): 03.10.2003*

UDK621.3:(53+54+621+66), ISSN0352-9045

SYSTEM DESIGN AND INTEGRATION IN PERVASIVE APPLIANCES

Manfred Glesner, Tudor Murgan, Leandro Soares Indrusiak,
Mihail Petrov and Sujan Pandey

Darmstadt University of Technology, Darmstadt, Germany

INVITED PAPER
MIDEM 2003 CONFERENCE
01. 10. 03 - 03. 10. 03, Grad Ptuj

Abstract: ubiquitous or pervasive computing environment needs to offer an important amount of essential features like proactivity, transparency, ease-of-use, high-level performance and energy management, cyber foraging and surrogate request support, location and context awareness, scalability, and so forth. Such environment-correlated attributes pose significant requirements on the employed hardware plat-forms. This works highlights the emerging hardware design paradigms supposed to comply with those strict requirements. Prior to this, we give a brief insight in the main evolutionary steps of pervasive computing, analyse the primary characteristics of calm technologies, and point out emerging hardware architectures and design methodologies. As a case study, an approach for the integration of reconfigurable hardware and computer applications is discussed.

Sistemsko načrtovanje in integracija v pervazivnih sistemih

Izvleček: Povsodno in prodorno računalniško okolje naj ponudi pomembno število bistvenih lastnosti, kot so proaktivnost, transparentnost, enostavnost uporabe, visok nivo storitev in energijskega obvladovanja, odvisnost od lokacije in vsebine itd. Takšni atributi odvisni od okolja postavljajo pomembne zahteve za delovanje strojnih platform. V prispevku opisujemo prihajajoče načrtovalske zglede, ki naj bi ugodili vsem tem strogim zahtevam. Pred tem podamo kratek pregled pomembnih razvojnih korakov prodornega računanja, analiziramo osnovne lastnosti novih tehnologij ter opozorimo na prihajajočo arhitekturo strojne opreme in načrtovalske metodologije. Kot primer smo prikazali pristop k integraciji rekonfiguracijske strojne opreme in računalniške aplikacije.

1. Introduction

Mark Weiser imagined the forthcoming ubiquitous computing systems as specialized elements of hardware and software, connected by means of both wired and wireless technologies /1/. Eventually, such elements should gracefully melt into the environment and become so ubiquitous that no one will notice their presence. By weaving themselves in an indistinguishable and diffuse fashion into the everyday life, pervasive technologies allow users to focus on tasks rather than tools /2/.

On one hand, as technology shrinks and the maximum die size enlarges, integrating complete systems of continuously increasing complexity becomes possible /3/. Moreover, new materials like organic semiconductors and plastic lasers, progress in communication technologies (especially wireless ones), as well as enhanced pattern recognition capabilities due to increasingly performant sensors offer the possibility to develop platforms and appliances only dreamt of a couple of decades ago.

On the other hand, technology improvements bring with them several challenges and drawbacks regarding increased delay, higher time-of-flight, strict requirements in clock and power distribution, higher noise, associated capacitive and inductive effects, increased leakage, relative power consumption, and heat dissipation /4/. In order to cope with those problems, a multitude of design paradigms like reconfigurable architectures, platform based design, IP reuse, orthogonalization of concerns, and communication abstraction emerged during the last years.

In a world composed of advanced communication components, highly sophisticated sensors, smart pens and tabs, such gadgets and the design thereof have to comply with a multitude of characteristics discussed in the sequel like proactivity, transparency, ease-of-use, energy management, cyber foraging, context awareness, scalability, and so forth.

2. Computing ages evolution

The term of ubiquitous computing, more recently also called pervasive computing, is subject to an impressive amount of redefinitions and variations and thus, a terminology labyrinth emerged. Terms like calm technology, ambient intelligence, proactive computing, invisible computing, disappearing computing, augmented reality, sentient computing, smart dust depict one and the same thing stated by Weiser in his seminal paper /1/: "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indis-*

tinguishable from it. ... A good tool is an invisible tool. ... The tool does not intrude on your consciousness; you focus on the task, not the tool."

2.1 The "Third Wave" of Computing

The first age of computing was the so-called *Main-frame Era*. At that time, each mainframe was shared by a group of people. The deserved group gradually reduced and a transition era marked by minicomputers fused with the second age of computing, that of the personal computer which allows a one-to-one connection user-PC. However, the PC is too attention demanding, too complex and hard to use by untrained or little trained users, isolating thus the user from people and activities. Consequently, the human became a peripheral of the computer and not vice-versa. Some drawbacks have been partly remedied through Internet and distributed computing. This second transition era is alleviating the evolution towards the third age of computing, the age of calm technology, the ubiquitous or pervasive computing era, in which the nowadays dominating desktop metaphor receives only a secondary importance and every person is served and attended by a gradually increasing number of embedded processors. Figure 1 depicts the evolution of the computing ages envisioned by Weiser.

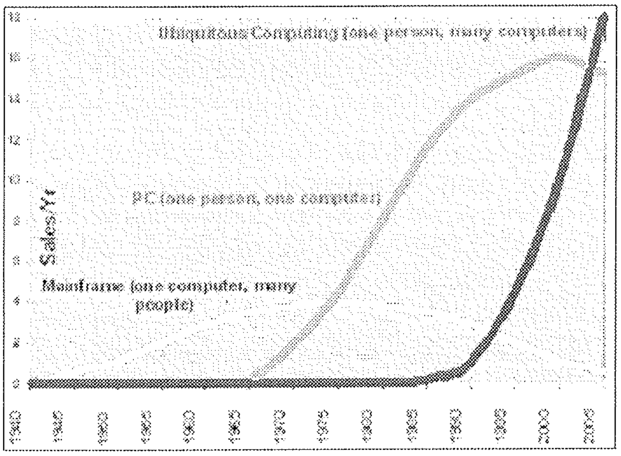


Figure 1: Evolution of computer ages (Source: Mark Weiser, http://www.ubicomp.com)

2.2 Related Research Fields

Satyanarayanan observed two distinct earlier steps in the evolution of pervasive computing, namely distributed systems and mobile computing /5/ as represented in figure 2. Thus, the technical problems induced by pervasive computing can be classified in two main categories: those related to the previous evolution steps which have been already identified - the solutions thereof can be easily mapped or adapted to the new needs; and those introduced by the ubiquitous computing paradigm requiring different solutions /2/.

Distributed systems emanated from the overlap of personal computers and local area network, belonging thus to the previously mentioned first transition era. Some of the fundamental issues addressed in this field are quintessential for pervasive computing: remote communication, fault tolerance in transactions, remote information access (distributed file systems and distributed databases), and security (authentication and privacy) /5/.

The requirements of distributed systems enhanced with the appearance of a new degree of freedom, i.e. mobile clients. Consequently, some key constraints posed by mobility arose: unpredictable network quality variation and local resource restriction entailed by weight, size, and battery power consumption constraints. The research issues in mobile computing have been focusing onto the following areas: mobile networking, mobile information access, adaptive application support, high-level energy saving techniques, and location sensitivity /5/.

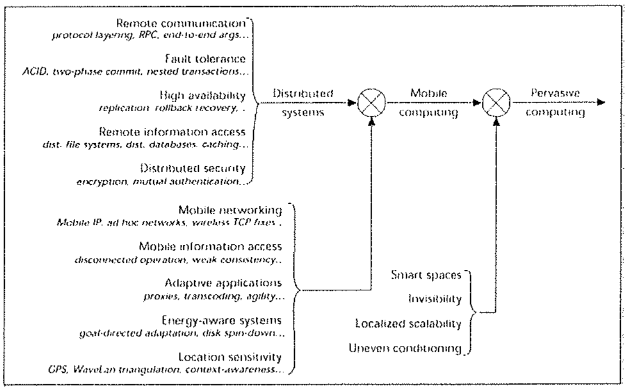

Remote communication
protocol layering, RPC, end-to-end args...
Fault tolerance
ACID, two-phase commit, nested transactions...
High availability
replication, rollback recovery, ..
Remote information access
dist. file systems, dist. databases, caching...
Distributed security
encryption, mutual authentication...
Distributed systems
Mobile computing
Pervasive computing
Mobile networking
Mobile IP, ad hoc networks, wireless TCP fixes ..
Mobile information access
disconnected operation, weak consistency..
Adaptive applications
proxies, transcoding, agility...
Energy-aware systems
goal-directed adaptation, disk spin-down...
Location sensitivity
GPS, WaveLan triangulation, context-awareness...
Smart spaces
Invisibility
Localized scalability
Uneven conditioning

Figure 2: Related fields to pervasive computing (Source: M. Satyanarayanan /5/)

For a user not being acutely aware of the employed technologies, pervasive computing needs to subsume mobile computing. Nevertheless, the emphasis of pervasive computing lies in four further additional research directions identified by Satyanarayanan /5/: smart space use, localized scalability, invisibility and uneven conditioning masking.

A smart space can be a local area network, an infrastructure-enhanced room, or even a body area network - the so-called wearable computing environment /6/. While such smart spaces enhance their functionality, the degree of interaction with users increases, and thus easy scalable spaces are required. Not only must scaling be invisible, but also must the use of such spaces be user transparent, or at least minimally distracting. Finally, masking uneven conditioning refers to reducing the amount of variation seen by the users. Thus, the personal computing space countervails against the low-ca'pable space.

3. System design challenges

Since the late 1980s, when the first ubiquitous computing project started in the Electronics and Imaging Laboratory of the Xerox Palo Alto Research Center /7/, various pervasive computing projects based on several scenarios emerged /8, 9, 10/. Early industrial ubiquitous computing projects included the Active Badge from Olivetti and later Xerox PARC, the Active Bat - an ultrasonic-based location system of AT&T's emerged out of Olivetti's project, the CoolTown Project of Hewlett Packard, the Pathfinder GPS watch of Casio, or Lancaster's handheld city tour guide system. Lately, numerous universities have also initiated pervasive computing projects.

3.1 Features and Requirements

Every user will be immersed in a personal computing space, which is accompanying her or him everywhere. This smart personal space, actually a set of interacting intelligent devices is an invisible interface with the surroundings. Such a personal space should support some missing features like proactivity, user-intent recognition, context-awareness, inter-layer knowledge exchange, and seamless task motion among platforms /5/.

In order to shift the desktop metaphor in the background, appropriate physical interactions have to be defined. Implicit inputs like GUIs are replaced or at least enhanced by handwriting, tactile, speech, and motion recognition. Data will be send to or acquired by a collection of devices and sensors that can process it in various ways. With an augmenting number of devices in a personal smart space, it gets increasingly difficult to provide ease-of-use. For an easy-to-use multi-modal user interface it is crucial to find the right balance between proactivity and transparency. A non proactive gadget will be "dumb", while a highly proactive one might get annoying.

A pervasive device entering into a smart space must be able to detect services and potential surrogates. Thus, surrogates - resources belonging to the infrastructure of the smart space - can temporarily assist a portable unit and dynamically augment the resources of a wireless device.

Open research points deal with discovery of and level-of-trust setting up with surrogates, balanced surrogate loading, low-intrusive integration of surrogates. Additionally, adaptation is highly necessary whenever major disequilibria between the demand and supply of resources appear.

Among other features, context awareness is fundamental for developing minimally intrusive pervasive appliances. Context awareness builds upon location awareness, a problem tackled for example by services as GPS.

Privacy and security are already painstaking problems in distributed systems. Furthermore, due to the intrinsic interaction of personal computing spaces with other personal spaces or the fixed infrastructure, those issues are of increased complexity in pervasive computing /5/.

Smart spaces can deploy themselves into user transparent or at least user-friendly environments if crucial features like reliability, availability, maintainability, and scalability are present /11/.

Last but not least, it is worth noticing that the overall power consumption increases because of proactivity and self-tuning. Due to the continuous pressure to make devices smaller, lighter, and more independent, stringent requirements on battery capacity is posed /2/. Low power circuitry is hereby insufficient, and high-level power management techniques like energy-aware tuning (applications switch to less power hungry modes of operation when possible) and memory management, or user intend deduction and remote task execution, i.e. the use of computational surrogates to increase battery life.

3.2 Impact on Layering

The impact on layering of pervasive appliances is of paramount importance. The merging of information from different layers for proactivity, adaptation, and dynamic power-performance management seem to require much more information exchange between layers in ubiquitous computing environments than in typical systems up to present /5/. The layer decomposition is a very tedious task, and the classical information hiding principle might not prove to be an efficient approach regarding ubiquitous computing.

Based on a simplified OSI layer model, one can generally talk about three super-layers /12/: the abstract layer - dealing with services, environment and soft-ware issues; the resource layer - comprising middle-ware, network control, and protocols; and the physical layer - focusing on flexible platforms, electronic design automation, low power issues.

Although the hardware appears to become more reliable, the software seems to grow more fragile while getting more powerful /2/. Furthermore, novel services have to be provided by an ubiquitous computing environment /13/. New services have to be informed about the local context of the user through a collection of sensors. Moreover, those services have to be enabled to affect the environment through a set of actuators.

In an intelligent space, communication and networking build the heart of the pervasive computing environment /14,15,16/.

Generally, the identified key communication challenges refer to the heterogeneity and topology of the networks, to short-lived and intermittent connectivity, and to the evolution and up-grade of long-lived systems /15/.

There are two fundamental approaches for designing a communication network, which must be carefully analysed for pervasive computing environments: infrastructured - with base stations, and infrastructure less - mobile ad-hoc networking. The latter seems to be very attractive for smart spaces, but nevertheless, the challenges regarding routing, security, reliability, QoS, and so on, increase in complexity.

4. Hardware design issues

As previously mentioned, pervasive computing is technology driven. For almost forty years, silicon integration has actually followed a rule that started as a speculation, i.e. Gordon Moore's Law /17/.

4.1 Orthogonalization of Concerns

Driven by the simultaneous demand of both performance and flexibility, platform-based design and reconfigurable architectures are getting an increasing interest in recent times /18,19/. Thus, a remarkable variety of different platforms have been proposed to trade energy-efficiency, cost and performance (see /20/ for a survey).

The SIA Roadmap /3/ predicts for the future 50 nm technology the integration of more than 4 billion transistors on a single chip running at 10 GHz and operating below 1V. Under these circumstances, one of the most important limiting factors for system performance, die area, and power consumption will be generated by the on-chip interconnect networks. Not only functional modules, or so-called IP blocks, have to be reused, but also the communication architecture and the interfaces between such blocks must be standardised. Therefore, orthogonalization of concerns /18/ and communication centric design /21/ emerged as possible solutions to fill the design gap.

Abstracting the physical interconnections through Networks-on-Chip (NoC) based design, will offer the possibility to cope with problems like distributed traffic monitoring and control, unavoidable data failures on the physical layer, synchronization, scalability, re-use, reliability, global asynchronous and local synchronous (GALS) communication /22/.

4.2 Interconnection Structures of Reconfigurable Architectures

Driven by the simultaneous demand of both performance and flexibility, reconfigurable architectures (see /20/ for a survey) are getting an increasing interest in recent days. For example, in the DSP context, the capacity of these devices to select the more appropriate data path for computing a task represents a significant advantage in terms of performance and power efficiency. Different platforms have been proposed to trade energy-efficiency, cost and performance. In /23/ a template for a heterogeneous reconfigurable architecture is proposed and used to instantiate a wireless base-band. A generic platform is presented in figure 3.

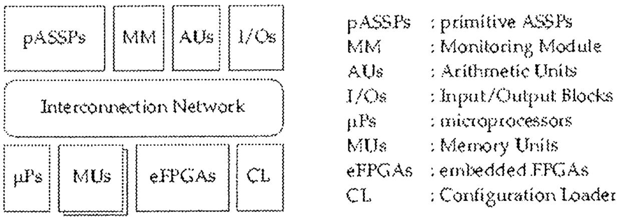


Figure 3: Reconfigurable template /30/

In both fine- and coarse-grained reconfigurable devices, the programmable interconnect architecture has a decisive influence in the total area, performance and power consumption of the system /24/.

To achieve high computational performance and flexibility, the different composing processing elements have to be richly interconnected. But since powerful interconnections imply large chip area and consume more energy, estimation procedures working at high level of abstraction are required to select the sufficient communication structure for a given application domain.

Several interconnection strategies for coarse-grained reconfigurable architectures have been reported. According to the degree of shareness, communication architectures can be classified into shared or dedicated structures. Dedicated architectures are generally composed by point-to-point like structures devoted to provide high performance communication of closed units. They use to be restricted to the first level of neighbours (e. g *DReAM*, *KressAray*) or they can include a second level as in *MATRIX*. The shared interconnection structures can be divided into three categories: multi-bus (e.g *RaPID*), crossbar (e.g. *PADDI-2*), and mesh; either regular (e.g. *DReAM*, *KressArray*, *MATRIX*, *Garp*) or irregular (*Pleiades*). A crossbar is the simplest way of interconnecting a given number of units and it guarantees full and arbitrary connectivity among elements. Thus, the mapping of any given network is attainable but the area requirement is of order O(n^2). Consequently, for large crossbars, wire size dominates chip area.

The mesh structure reduces the total number of switches needed by limiting the connections to fixed distances. This type of network requires a smaller amount of area, but it makes distant communication slower and more expensive. This enforces a detailed performance analysis of the architecture before implementation /25/.

Another important communication issue is the connection between the reconfigurable computing elements and the host computer. There are two main approaches: either the reconfigurable unit is integrated into the processor core as a functional unit (e.g. *PRISC*, CHIMAERA); or the reconfigurable unit is placed as a coprocessor close to the host unit, sharing the cache unit (e. g *Garp*, *REMARC*, *MorphoSys*). Although theoretically, it is also possible to place the reconfigurable unit as an attached processing unit outside the processor, the low performance of this solution made this solution impractical.

Further on, in /26/ the possibility of integrating switching network concepts within a reconfigurable architecture has been analysed. At the physical layer, the communication architecture is based on a simple point-to-point structure conforming a 2-D mesh of routers. The functionality of the data-link and network layers defines a packet protocol that can be used as a virtual connection between all the processing units of the configurable architecture.

4.3 Power Estimation and Optimisation

In portable appliances, batteries are a significant source of size, weight, and mechanical inflexibility /27/. There is still a significant need to improve and optimise batteries even though the energy of the human body is envisaged to be used for personal computing spaces. Nevertheless, power consumption must be reduced due to rapidly increasing thermal dissipation issues (packaging costs).

Benefiting from the non-uniformity of the workload in various signal processing applications, several dynamic power management policies have been developed /28/. Nevertheless, the integration of on-line power, performance and information-flow management strategies based on traffic monitoring in (dynamically) reconfigurable templates has yet to be explicitly tackled /29/. Similar strategies will be an in-separable part of pervasive devices /30/.

A hierarchy of increasing complexity stochastic data models that effectively exploits the knowledge about the excitation of the system was developed in /31/ order to estimate the power consumption in digital circuits by modelling the effects of the high level signal characteristics on the power consumption. The hierarchy of models includes a Gaussian ARMA model for linear systems, a cyclic multiplexing of Gaussians for folded and multiplexed architectures, a general uncorrelated model for memoryless non-linear architectures, and a general correlated model for non-linear circuits with reconvergent paths. Such techniques can also be integrated in the design flow of reconfigurable platforms.

5. Ubiquitous access to reconfigurable hardware

An approach for the integration of reconfigurable hardware and computer applications based on the concept of ubiquitous computing is presented in /32/. The goal is to allow a network of reconfigurable hardware modules to be transparently accessible by client applications. The communication between them is done at the API level, and a Jini-based infrastructure is used to provide an interface for the client applications to find available reconfigurable hardware modules over the network. A DES-based cryptography system was implemented as a case study.

The aim is to reduce the integration overhead of reconfigurable hardware modules and computer systems. Such overhead can be reduced by raising the level of abstraction of the integration architecture, allowing the communication to be done via message passing, as proposed in the object-oriented paradigm. By using this approach, each reconfigurable hardware module can be seen by the rest of the system as an object. Thus, it should be reconfigured and used through method calls. This can make a significant difference for the system designer, because he/she can abstract the internal details of the reconfigurable module - a typical result of the encapsulation feature of object-oriented systems - and can design the whole system communication at the API level. In such approach, all the subsystems depending on the reconfigurable hardware module can call a configuration method to set up the desired functionality, and then call methods to pass the data to be processed and receive the results. Figure 4 depicts such possibility.

In order to cope with the demands of the current application scenarios - where the computation is performed by several interconnected appliances - one also has to support the integration of reconfigurable hardware modules into distributed computer systems. We can expect that each subsystem could be in a different location, connected to the others by a heterogeneous network. So, the approach should allow those subsystems to interact with any number of encapsulated reconfigurable hardware modules. The minimum infrastructure to do so comprehends distributed resource localization and remote method invocation. The first technique provides means for the distributed objects to locate other objects according to their needs, while the second one is responsible for the common dialect used by the objects to exchange messages once they have established communication.

Many of the applications of reconfigurable hardware can benefit from the proposed approach. For instance devices, which were already deployed - such as an ad-hoc network of sensors - could be located and upgraded by method calls if they have encapsulated reconfigurable hardware supported by an infrastructure for localization and remote method invocation. Another application scenario would be the use of reconfigurable hardware modules as accelerators for specific computational tasks. For instance, a mobile device, which needs to decode a stream of data and does not have the computational power to do so could use the resource localization feature to search for a reconfigurable hardware module which is able to decode the data stream.

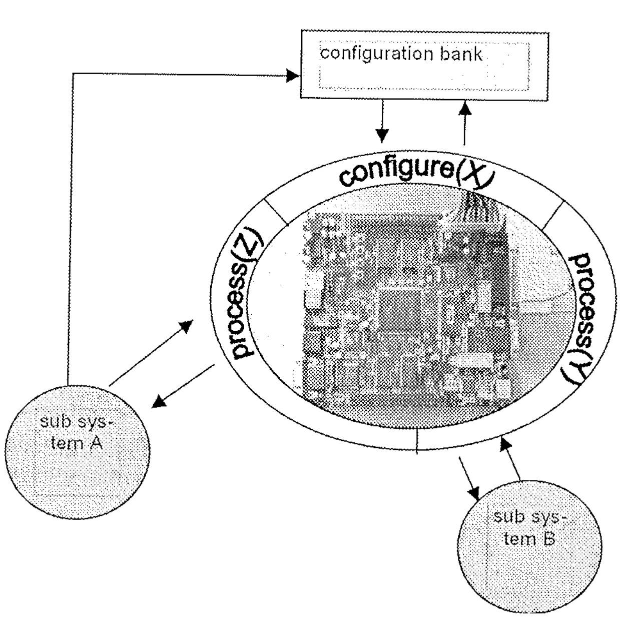


Figure 4: Reconfigurable hardware encapsulation

A third application could be found on the use of reconfigurable hardware as a prototyping platform. Let's imagine a system design specification done in the functional level. Once that specification fulfils the functional requirements, it should be submitted to successive synthesis steps in order to be implemented as a physical entity. Reconfigurable platforms are often used as an intermediate stage within such process, allowing system designers to verify the correctness of their designs prior to the final implementation. Our approach could provide a simpler way to integrate the functional specification with the prototyping platform, in such a way that they can interoperate. This would allow a mixture of simulation and emulation in the functional level, because one could synthesize and implement part of the functional specification in the reconfigurable hardware and still be able to perform the functional simulation, as the rest of the specification would communicate with the prototype in the same way it did before with the functional description.

Similarly to the prototyping platform scenario, a distributed IP core validation system could benefit from the proposed approach. In simulation systems a designer access remotely an IP core so it can be simulated together with the rest of the design. The approach could provide a layer between the IP core repository and the client, so the cores could be accessed seamlessly, without a previous connection to a predefined server. Another advantage would be the possibility of simulating an actual core implemented in a reconfigurable module, instead of the simulation models.

6. Conclusions

This work discussed the main features of pervasive appliances, described the evolution of computing ages, and highlighted the primary requirements and constraints posed especially on the hardware platforms and the design thereof.

Moreover, an approach for the integration of reconfigurable hardware and computer applications based on the concept of ubiquitous computing was discussed. The goal was to reduce the integration overhead of reconfigurable hardware modules and computer systems.

Even though pervasive computing is rather driven by technology, it is not to be successful only if technology does, but also if it can support and ease social life. By becoming a larger part of human existence, pervasive devices might promote but also inhibit social relationships /33/, thus system designers should be ready to deal additionally with such non-technical constraints.

7. References

/1/ Weiser, M.: The Computer for the 21st Century. *Scientific American*, 265(3):66-75, September 1991

/2/ Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. *Communi-cations of the ACM*, 36(7):74-84, July 1993

/3/ Allan, A.; Edenfeld, D.; Joyner, W. H. Jr.; Kahng, A. B.; Zorian, Y.: 2001 Technology Roadmap for Semiconductors. *IEEE Computer*, 35(1):42-53, January 2002

/4/ Sylvester, D.; Keutzer, K.: A global wiring paradigm for deep submicron design. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pages 242-252, February 2000

/5/ Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, August 2001

/6/ Starner, T.: Wearable Computing: No Longer Science Fiction. *IEEE Pervasive Computing*, pages 86-88, January-March 2002

/7/ Weiser, M.; Gold, R.; Brown, J. S.: The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems Journal*, 38(4): 693-696, 1999

/8/ Want, R.; Schilit, B: Expanding the Horizons of Location-Aware Computing. *IEEE Computer*, pages: 1-4, August 2001

/9/ Want, R.; Pering, T.; Boriello, G.; Farkas, K.: Disappearing Hardware. *IEEE Pervasive Computing*, pages 36-47, January-March 2002

/10/ Abowd, G. D.; Mynatt, E. D.; Rodden, T.: The Human Experience. *IEEE Pervasive Computing*, pages: 48-57, January-March 2002

/11/ Hennessy, J.: The Future of Systems Research. *IEEE Computer*, pages: 27-33, August 1999

/12/ Ciarletta, L.; Dima, A.: A Conceptual Model for Pervasive Computing. Intl. Workshop on Parallel Processing, Toronto, Canada, 21-24 August 2000

/13/ Kindberg, T; Fox, A.: System Software for Ubiquitous Computing. *IEEE Pervasive Computing,* pages: 70-81, January-March 2002

/14/ Boriello, G.: The Challenges to Invisible Computing. *IEEE Computer*, pages: 123-125, November 2000

/15/ Boriello, G.: Key Challenges in Communication for Ubiquitous Computing. *IEEE Communications Magazine*, pages: 16-18, May 2002

/16/ Estrin, D; Culler, D.; Pister, K.; Sukhatme, G.: Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Computing*, pages: 59-69, January-March 2002

/17/ Boekhorst, F.: Ambient intelligence, the next paradigm for consumer electronics: how will it affect silicon? IEEE International Solid-State Circuits Conference. Digest of Technical Papers. pages: 28 -31, vol.1, 3-7 Feb. 2002

/18/ Keutzer, K; Newton, A. R.; Rabaey, J. M.; Sangiovanni-Vincentelli, A.: System-Level Design: Orthogonalization of Concerns and Platform-Based Design. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 19(12): 1523-1543, December 2000

/19/ Rabaey. J.: Reconfigurable Processing: The Solution to Low-Power Programmable DSP, In Proc. of the IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, pages 275-278, April 1997

/20/ Hartenstein, R.: A Decade of Reconfigurable Computing: A Visionary Retrospec-tive. In Proc. of the DATE Conference, pages: 642-649, 2001

/21/ Sgroi, M.; Sheets, M.; Mihal, A.; Keutzer, K.; Malik, S.; Rabaey, J.; Sangiovanni-Vincentelli, A.: Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based De-sign, In Proc. of the DAC, pages: 667-672, 2000

/22/ Benini, L.; De Micheli, G.: Networks on Chips: A New SoC Paradigm. IEEE Computer, pages: 70-78, January 2002

/23/ Zhang, H.; Prabhu, V.; Rabaey, J.; et al. A 1-V heterogeneous reconfigurable DSP IC for wireless baseband digital signal processing. *Journal of solid state circuits*, pages 1697 - 1704, November 2000.

/24/ Rabaey, J.; Reconfigurable Processing: The Solution to Low-Power Programmable DSP. In IEEE Intl. Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 275-278, April 1997

/25/ Zhang, H.; Wan, M.; George, V.; Rabaey, J ; Interconnect Architecture Exploration for Low-Energy Reconfigurable Single-Chip DSPs, IEEE Computer Society Workshop on VLSI'99, April 1999, Orlando, Florida

/26/ Michael B. Taylor, Jason Kim, et al. The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *IEEE Micro*, 22, March-April 2002.

/27/ Hahn, R.; Reichl, H.: Batteries and power sup-plies for wearable and ubiquitous computing, The Third International Symposium on Wearable Computers, Digest of Papers, pages: 168 - 169, 18-19 Oct. 1999

/28/ Benini, L.; De Micheli, G.: System-Level Power Optimization: Techniques and Tools, *ACM Trans. on Design Automation of Electronic Systems*, 5(2): 115-192, April 2000

/29/ Benini, L.; De Micheli, G.: Powering Networks on Chips: Energy efficient and reliable interconnect design for SoCs. In Proc. of the 14th Inter-national Symposium on System Synthesis, pages: 33-38, 30 Sept.-3 Oct. 2001

/30/ Murgan, T; Petrov, M; Garcia Ortiz, A.; Ludewig, R.; Zipf, P.; Hollstein, T.; Glesner, M.; Oelkrug, B.; Brakensiek, J; Evaluation and Run-Time Optimisation of On-Chip Communication Structures in Reconfigurable Architectures, In Proc. of the Intl. Conference on Field Programmable Logic and Applications, Lisbon, Portugal, 1-3 September 2003

/31/ García Ortiz, A.; Stochastic data models for power estimation at high-levels of abstraction, Ph. D. Thesis, Darmstadt University of Technology, June 2003

/32/ Indrusiak, L.; Lubitz, F.; Reis, R.; Glesner, M.; Ubiquitous access to reconfigurable hardware: application scenarios and implementation issues. In Proc. of the Design, Automation and Test in Europe Conference and Exhibition, pages: 940-945, March 3-7, 2003

/33/ Dryer, D. C.; Eisbach, C.; Ark, W. S.: At what cost pervasive? A social computing view of mobile computing systems. *IBM Systems Journal*, 38(4): 652-676, 1999

Manfred Glesner, Tudor Murgan, Leandro Soares
Indrusiak, Mihail Petrov and Sujan Pandey
Darmstadt University of Technology
Karlstr. 15, D-64283 Darmstadt, Germany
{glesner, murgan, lsi, pmihail, spandey}
@mes.tu-darmstadt.de

Prispelo (Arrived): 15.09.2003   Sprejeto (Accepted): 03.10.2003

CONFERENCE MIDEM 2003 - REPORT

The 39th **International Conference on Microelectronics, Devices and Materials, MIDEM 2003,** was held in Ptuj, from the first to the third of October 2003. Wonderful architecture of the castle and pleasant conference accompanying events added a special touch to the already very interesting presentations.

This conference continued the tradition of annual international conferences organised by MIDEM, Society for Microelectronics, Devices and Materials, Ljubljana, Slovenia.

49 papers and nine invited presentations in six sessions and in included workshop on Embedded Systems was presented during three days from Wednesday to Friday. The presentations at the Conference were grouped in the following Sessions: Ceramics Metals and Composites; Integrated Circuits; Sensors; Optoelectronics; Device Physics and Modelling, and Device Physics, Modeling and Technology.

All invited papers are presented in this issue of the Journal.

This year, the workshop was focused on the embedded systems which are rapidly becoming one of the driving factors in the electronic industry. The workshop covered a broad scope of embedded systems, including embedded systems architecture, systems-on-chip (SoC) embedded systems, embedded systems design cases and applications, current trends in embedded systems configurability, embedded systems testing and software development cycle. It is important to note, that in addition to five distinguished lecturers from the abroad, several researchers from the national electronic and telecommunications industry presented their latest results and achievements, and thus stimulated the discussion with real industrial case studies. The workshop was organized by the Electronics Department of the Faculty of Electrical Engineering, University of Ljubljana.

We hope that you will be able to remember this event not only for the importance of the papers and discussions, but also for the many new friendships and pleasant memories of the country.

	NAME	COMPANY - INSTITUTION	ADDRESS	P. CODE	CITY	
1	AKIL MOHAMED	ESIETE	2 BLVD.PASCAL BP GG	93162	NOISY LE GRAND	F
2	ALJANČIČ UROŠ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
3	AMON SLAVKO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
4	ARTIČEK JURE	ISKRAEMECO	SAVSKA LOKA 4	4000	KRANJ	SLO
5	BECKER JUERGEN	KARLSRUHE UNIVERSITY	D-76128 KARLSRUGE	76128	KARLSRUHE	D
6	BELAVIČ DARKO	HIPOT-HYB d.o.o	TRUBARJEVA 7	8310	ŠENTJERNEJ	SLO
7	BERNIK SLAVKO	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
8	BOCK HOLGER	INFINEON TECHNOLOGIES	BABENBERGERSTR. 10	9500	VILLACH	A
9	BRECL KRISTIJAN	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
10	CVIKL BRUNO	GRADBENA UNIVERZA V MARIBORU	SMETANOVA 17	2000	MARIBOR	SLO
11	ČADEŽ BORUT	ISKRATEL	LJUBLJANSKA C.24A	4000	KRANJ	SLO
12	ČAKARE SAMARDŽIJA LAILA	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
13	ČELAN ŠTEFAN	MESTNA OBČINA PTUJ	MESTNI TRG 1	2250	PTUJ	SLO
14	DEDIČ JOŽE	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
15	DRAŽIČ GORAN	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
16	FETIH MATEJ	ISKRAEMECO	SAVSKA LOKA 4	4000	KRANJ	SLO
17	FROECHLICH HUBERT	ISKRA TRANSMISSION	STEGNE 11	1000	LJUBLJANA	SLO
18	GLAŽAR BOŠTJAN	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO

19	GRADIŠNIK VERA	SVEUČILIŠTE U RIJECI	PRIMORSKA 42	51410	OPATIJA	CRO
20	GRAMC JANEZ	HIPOT-HYB d.o.o	TRUBARJEVA 7	8310	ŠENTJERNEJ	SLO
21	GRUDEN STANISLAV	ISKRAEMECO	SAVSKA LOKA 4	4000	KRANJ	SLO
22	HARTENSTEIN REINER	UNIV. KAISERSLAUTERN	POSTFACH 1744	76607	KAISERSLAUTERN	D
23	HROVAT MARKO	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
24	JANKOVEC MARKO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
25	JOVANOVIĆ VLADIMIR	FAKULTETA ZA ELEKTROTEHNIKO	UNSKA 3	10000	ZAGREB	CRO
26	KORIČIĆ MARKO	FAKULTETA ZA ELEKTROTEHNIKO	UNSKA 3	10000	ZAGREB	CRO
27	KOROŠAK DEAN	FAKULTETA ZA GRADBENIŠTVO	SMETANOVA 17	2000	MARIBOR	SLO
28	KOSEC MARIJA	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
29	KOŽELJ MATJAŽ	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
30	KRAMER JANJA	FAKULTETA ZA GRADBENIŠTVO	SMETANOVA 17	2000	MARIBOR	SLO
31	KRČ JANEZ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
32	KRSNIK ZORAN	ISKRAEMECO	SAVSKA LOKA 4	4000	KRANJ	SLO
33	LEY MANFRED	CTI	EUROPASTR. 4	9500	VILLACH	A
34	MAČEK SREČKO	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
35	MERC UROŠ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
36	MOZETIČ MIRAN	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
37	MOŽEK MATEJ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
38	MURGAN TUDOR	DARMSTADT UNIV.OF TECHNOLOGY	KARLSTR.15	64283	DARMSTADT	D
39	NOVAK FRANC	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
40	PAVLIN MARKO	HIPOT-HYB d.o.o	TRUBARJEVA 7	8310	ŠENTJERNEJ	SLO
41	PERNE JOŽEF	ISKRAEMECO	SAVSKA LOKA 4	4000	KRANJ	SLO
42	PIGNATEL GIORGO	UNIVERSITY OF PERUGIA	VIA G. DURANTI 93	6125	PERUGIA	ITA
43	PINTERIČ MARKO	FAKULTETA ZA GRADBENIŠTVO	SMETANOVA 17	2000	MARIBOR	SLO
44	PLETERŠEK ANTON	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
45	POLANŠEK GREGOR	ISKRATEL	LJUBLJANSKA C.24A	4000	KRANJ	SLO
46	RAIČ DUŠAN	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
47	RESNIK DRAGO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
48	ROČAK DUBRAVKA	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
49	ROJAC TADEJ	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
50	SANTO ZARNIK MARINA	INSTITUT JOZEF STEFAN	JAMOVA 39	1000	LJUBLJANA	SLO
51	SEDLAKOVA VLASTA	BRNO UNIVERSITY OF TECHNOLOGY	TECHNICKA 8	61600	BRNO	CZ
52	SERNEC RADOVAN	IPS	C.LJUBLJANSKE BRIGADE 17	1000	LJUBLJANA	SLO
53	SEŠEK JANEZ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
54	SIKULA JOSEF	BRNO UNIVERSITY OF TECHNOLOGY	TECHNICKA 8	61600	BRNO	CZ

55	SITEK JANUSZ	TELE AND RADIO RESEARCH INSTITUTE	RATUSZOWA 11	3450	WARSZAW	PL
56	SMOLE FRANC	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
57	STARAŠINIČ SLAVKO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
58	STEPANIĆ JOSIP	FACULTY OF MECHANICAL ENG. AND NAVAL ARCHITECTURE	I.LUČIĆA 5	10000	TAGREB	CRO
59	STRLE DRAGO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
60	SUHADOLNIK ALOJZ	FAKULTETA ZA STROJNIŠTVO	AŠKRČEVA 6	1000	LJUBLJANA	SLO
61	ŠTEMBERGER IGOR	ISKRA TRANSMISSION	STEGNE 11	1000	LJUBLJANA	SLO
62	ŠTERN ANTON	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
63	TEŽAK OTO	FAKULTETA ZA ELEKTROTEHNIKO IN RAČUNALNIŠTVO	SMETANOVA 17	2000	MARIBOR	SLO
64	TOPIČ MARKO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
65	TRONTELJ JANEZ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
66	TRONTELJ JANEZ JR.	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
67	TROST ANDREJ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
68	VRTAČNIK DANILO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
69	VUKADINOVIČ MIŠO	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
70	ZUPANČIČ SREČKO	ISKRATEL	LJUBLJANSKA C.24A	4000	KRANJ	SLO
71	ŽEMVA ANDREJ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO
72	ŽUNIČ ANDREJ	FAKULTETA ZA ELEKTROTEHNIKO	TRŽAŠKA 25	1000	LJUBLJANA	SLO

Informacije MIDEM
Strokovna revija za mikroelektroniko, elektronske sestavine dele in materiale

NAVODILA AVTORJEM

Informacije MIDEM je znanstveno-strokovno-društvena publikacija Strokovnega društva za mikroelektroniko, elektronske sestavne dele in materiale - MIDEM. Revija objavlja prispevke s področja mikroelektronike, elektronskih sestavnih delov in materialov. Ob oddaji člankov morajo avtorji predlagati uredništvu razvrstitev dela v skladu s tipologijo za vodenje bibliografij v okviru sistema COBISS.
Znanstveni in strokovni prispevki bodo recenzirani.

Znanstveno-strokovni prispevki morajo biti pripravljeni na naslednji način:

1. Naslov dela, imena in priimki avtorjev brez titul, imena institucij in firm
2. Ključne besede in povzetek (največ 250 besed).
3. Naslov dela v angleščini.
4. Ključne besede v angleščini (Key words) in podaljšani povzetek (Extended Abstract) v anglešččini, če je članek napisan v slovenščini
5. Uvod, glavni del, zaključek, zahvale, dodatki in literatura v skladu z IMRAD shemo (Introduction, Methods, Results And Discsussion).
6. Polna imena in priimki avtorjev s titulami, naslovi institucij in firm, v katerih so zaposleni ter tel./Fax/Email podatki.
7. Prispevki naj bodo oblikovani enostransko na A4 straneh v enem stolpcu z dvojnim razmikom, velikost črk namanj 12pt. Priporočena dolžina članka je 12-15 strani brez slik.

Ostali prispevki, kot so poljudni cčlanki, aplikacijski članki, novice iz stroke, vesti iz delovnih organizacij, inštitutov in fakultet, obvestila o akcijah društva MIDEM in njegovih članov ter drugi prispevki so dobrodošli.

Ostala splošna navodila

1. V članku je potrebno uporabljati SI sistem enot oz. v oklepaju navesti alternativne enote.
2. Risbe je potrebno izdelati ali iztiskati na belem papirju. Širina risb naj bo do 7.5 oz.15 cm. Vsaka risba, tabela ali fotografija naj ima številko in podnapis, ki označuje njeno vsebino. Risb, tabel in fotografij ni potrebno lepiti med tekst, ampak jih je potrebno ločeno priložiti članku. V tekstu je treba označiti mesto, kjer jih je potrebno vstaviti.
3. Delo je lahko napisano in bo objavljeno v slovenščini ali v angleščini.
4. Uredniški odbor ne bo sprejel strokovnih prispevkov, ki ne bodo poslani v dveh izvodih skupaj z elektronsko verzijo prispevka na disketi ali zgoščenki v formatih ASCII ali Word for Windows. Grafične datoteke naj bodo priložene ločeno in so lahko v formatu TIFF, EPS, JPEG, VMF ali GIF.
5. Avtorji so v celoti odgovorni za vsebino objavljenega sestavka.

Rokopisov ne vračamo. Rokopise pošljite na spodnji naslov.

Uredništvo Informacije MIDEM
MIDEM pri MIKROIKS
Stegne 11, 1521 Ljubljana, Slovenia
Email: Iztok.Sorli@guest.arnes.si
tel. (01) 5133 768, fax. (01) 5133 771

Informacije MIDEM
Journal of Microelectronics, Electronic Components and Materials

INSTRUCTIONS FOR AUTHORS

Informacije MIDEM is a scientific-professional-social publication of Professional Society for Microelectronics, Electronic Components and Materials - MIDEM. In the Journal, scientific and professional contributions are published covering the field of microelectronics, electronic components and materials. Authors should suggest to the Editorial board the classification of their contribution such as : original scientific paper, review scientific paper, professional paper...
Scientific and professional papers are subject to review.

Each scientific contribution should include the following:

1. Title of the paper, authors' names, name of the institution/company.
2. Key Words (5-10 words) and Abstract (200-250 words), stating how the work advances state of the art in the field.
3. Introduction, main text, conclusion, acknowledgements, appendix and references following the IMRAD scheme (Introduction, Methods, Results And Discsussion).
4. Full authors' names, titles and complete company/institution address, including Tel./Fax/Email.
5. Manuscripts should be typed double-spaced on one side of A4 page format in font size 12pt. Recommended length of manuscript (figures not included) is 12-15 pages
6. Slovene authors writing in English language must submit title, key words and abstract also in Slovene language.
7. Authors writing in Slovene language must submit title, key words and extended abstract (500-700 words) also in English language.

Other types of contributions such as popular papers, application papers, scientific news, news from companies, institutes and universities, reports on actions of MIDEM Society and its members as well as other relevant contributions, of appropriate length , are also welcome.

General informations

1. Authors should use SI units and provide alternative units in parentheses wherever necessary.
2. Illustrations should be in black on white paper. Their width should be up to 7.5 or 15 cm. Each illustration, table or photograph should be numbered and with legend added. Illustrations, tables and photographs must not be included in the text but added separately. However, their position in the text should be clearly marked.
3. Contributions may be written and will be published in Slovene or English language.
4. Authors must send two hard copies of the complete contributon, together with all files on diskette or CD, in ASCII or Word for Windows format. Graphic files must be added separately and may be in TIFF, EPS, JPEG, VMF or GIF format.
5. Authors are fully responsible for the content of the paper.

Contributions are to be sent to the address below.

Uredništvo Informacije MIDEM
MIDEM pri MIKROIKS
Stegne 11, 1521 Ljubljana, Slovenia
Email: Iztok.Sorli@guest.arnes.si
tel.+386 1 5133 768, fax.+386 1 5133 771