

RRL - AN INTEGRATED ENVIRONMENT OF ROBOT PROGRAMMING

INFORMATICA 1/92

Keywords: robot programming, off-line programming, robot controller, CAD systems

Bojan Nemec¹, Anton Ružič¹, Vinko Ilc²
¹Jozef Stefan Institute, University of Ljubljana
²Riko - Ribnica

The paper describes RRL, an integrated robot programming environment, which includes a robot controller with a dialogue-oriented robot monitor, editor, robot language interpreter and an additional computer for graphic simulation and off-line program development. Trajectories can be generated by a CAD system, verified on the off-line programming system and then transferred and executed on the real robot. One of the main advantages of the system is that it can be implemented on a low cost-hardware. Off-line programming uses the identical trajectory generation module as the robot controller and includes simulation of the robot dynamics. Therefore, the simulation gives extremely credible results. The system described is in use in several industrial applications.

RRL - integrirano okolje za programiranje robotov:

Članek opisuje programski sistem RRL, ki vključuje programsko opremo na robotskem krmilniku in programsko opremo na zunanjem računalniku za programiranje z simulacijo. Trajektorije, ki sestavljajo delavno nalogo, robota lahko določimo s pomočjo CAD sistema, preiskusimo z simulacijo in prenesemo v robotski krmilnik. Prednost predstavljenega sistema sistema je tudi v tem, da je implementiran tudi na ceneni PC računalnikih.

1 Introduction

Unlike in NC machine programming, there is no industrial standard for robot programming. Almost each robot manufacturer offers his own robot programming language [1,2,3]. Textual robot programming languages offer many advantages, but the teach-in principle is still widely used for robot programming [1,2,3]. The main reasons why the teach-in principle is still used are the following:

- simple operation; the operator does not have to learn any programming language.
- teaching by showing nearly eliminates all logical errors in trajectory definition.
- simple debugging and correction of the robot program.

debugging and correction of robot programs can be efficiently done by off-line development and ver-

ification via graphic simulation. Additionally, off-line programming supported by CAD allows work cell analysis in the early stage of work cell planning [4]. In the past, many off-line programming systems were developed [5,6,7,8], but there are some problems which limit the use of off-line robot programming, such as:

- Post processors are not available only for all robots or robot programming languages.
- If the simulation software does not include the same type of trajectory generation algorithms, the simulation will give false results, especially in cycle time calculation.
- Most of the off-line simulation packages do not include the dynamics of the robot, hence it is always questionable if the simulation will be close enough to the behaviour of the real robot, especially at high speed.

- off-line programming often requires expensive hardware and software which may surpass the cost of the robot itself.

For the new generation of RIKO industrial robots, we developed a programming environment called RRL (Riko Robot Language) which tries to solve the above-mentioned problems. The RRL program environment has the versatility of textual programming languages and ease of use and debugging capabilities of teach-in programming. RRL allows off-line program development and graphic simulation which runs on a low cost PC computer, coupled with a robot controller.

2 Basic structure of RRL

The RRL robot program environment consists of two main modules:

- RRL program environment on the robot controller, which allows on-line program development and execution.
- RRL program environment on the host computer, which allows off-line program development and simulation.

2.1 RRL program environment on the robot controller

RRL on the robot controller consists of the following modules (Fig 1) :

- PasRo kernel. Pasro is a set of pascal procedures and functions for robot programming [9]. It is a powerful robot programming language by itself, but requires knowledge of Pascal programming and a Pascal compiler to run. In our case, PasRo is used as a kernel for RRL development.
- RRL interpreter is the interpreter of a high level, motion-oriented robot programming language.
- RRL editor is a menu-oriented, screen editor. It is designed to simplify the writing and editing of the textual part of the RRL program. A syntax check is performed at this level and therefore it is impossible to write a syntactically incorrect program.

- RRL frames definition module consists of a frame editor where frames are entered or edited explicitly using the main control panel and the Teach-in module, where frames are defined implicitly using the teach panel.

- RRL monitor, which links other modules and allows saving and loading of robot programs. Up to 42 programs can be stored the RRL directory. There are several modes of execution of the robot program :

- continuous execution of the program,
- step-by-step execution,
- continuation of execution of the program. The program can be changed and then continued from every step.
- step-by-step execution of the last 32 movements in the reverse order.

- RRL computer link module allows connection of the robot controller to the host computer. The host computer can send or receive any program or parameter field from or to the robot controller. Additionally, it can take control of the execution of the program and send commands directly to the robot or receive data from the robot. The entire programming system is menu-oriented. The user does not have to type or remember all commands but simply selects the required command from the menu.

2.2 Robot language description

The RRL is designed as a compromise between simplicity of use and ability to program complex tasks. The main demands in developing RRL were:

- to be simple to understand and simple to learn,
- to allow easy menu oriented programming,
- to be easily expandable with new instructions.

RRL can drive a robot with up to 6 robot axes and up to 3 external axes and can coordinate robot trajectory with up to 3 arbitrarily positioned external axes. RRL can be used to program most of the

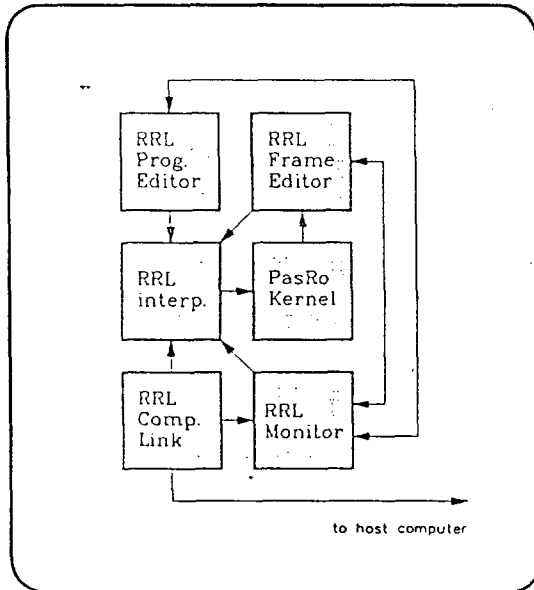


Figure 1: Structure of the RRL system and data flow on the robot controller

tasks which can be performed by the RIKO 106 welding/assembly robot. For very complex tasks with intensive sensor interaction, we can develop a program on the host computer in PasRo and control the robot with the host computer through the Computer Link.

RRL contains instructions for :

- different types of interpolations

| | |
|--------|--|
| JMOVE | movement with joint interpolation |
| SMOVE | straight move in Cartesian coordinates |
| VMOVE | SMOVE via intermediate frames without stopping |
| CIRCLE | circular interpolation |
| SPLINE | spline interpolation through a set of frames |
| LINE | straight move with selectable velocity profile |
| APPRO | approach to the point with joint interpolation |
| DEPART | straight relative movement |
| DRIVE | relative movements of the joint |
| FTRACK | activating of hybrid force position control using up to 6 d.o.f force/torque sensor. |

All movement instructions except LINE generates a smooth trajectory using 4-1-4 splines

[10].

- instructions for movement definition

| | |
|---------|---|
| MAXSP | definition of the max. speed in mm/s and dg/sec |
| SPEED | relative speed in % of maximum speed |
| SPPR | selection of the speed priority |
| MOVSEL, | selection and definition of the coordination |
| SYNMO | between robot axes and external axes |
| ACC | selection of the acceleration factor |
| TCP | tool geometry definition. Up to 10 tools with tcp vector and orientation angles can be defined on line shift and rotation of the frames |
| SHIFT | |
| PALLET | definition for manipulation with up to 10 pallets |

- frame arithmetics

| | |
|-------|---|
| FRAME | assignment, addition, subtraction and rotation of the frames. |
|-------|---|

- instruction for general program facilities

| | |
|---------------|--|
| SET | assignment to the integer value |
| GOTO, CALL | unconditional branching and calling a subprogram |
| IF THEN ENDIF | conditional branching |
| WRITE | output on the user window of the screen |
| CHAIN | chain to another robot program |
| DELAY, STOP | |

- instructions for process synchronization

| | |
|----------------|--|
| IFS THEN ENDIF | conditional branching on input signal status |
| SIGON, SIGOF, | activating of an output signal |
| PULSE | |
| WAIT | wait for the defined event |

- instructions for arc welding

| | |
|--------------|--|
| WLDST, WLDSP | start and stop of arc welding |
| WLDCL | cleaning of welding pistol |
| WEAVE | weaving with different patterns during arc welding |
| WPAR | definition of welding parameters |

2.3 RRL programing environment for the host computer

The basic structure of the RRL off-line programming environment is presented in Fig 2. and consists of the following modules :

- PasRo kernel, which is identical to the kernel of the RRL kernel for the robot controller
- RRL interpreter, which is identical to the interpreter of the robot controller
- RRL program file and location definition file compiler, which compiles the source code for the textual program and file definition into the internal format of the robot controller.
- Align module, which translates and rotates the whole location definition file to the correct position, using three reference points
- Any text editor
- CAD package with the capability of defining 3-D wireframe objects (e.g. AutoCad Version 9.0 or later)
- Simulation system module, which simulates the kinematics and dynamics of the robot and displays the simulated cell on the graphic terminal. Although the simulation module is primarily intended for the Riko-106 6 axis electrical robot, it can be user-adapted to other robots with similar kinematic configurations.
- Module for converting DXF drawing format to the internal format of the simulation system. The user can define or modify the robot or cell components using a CAD package. DXF file format is then converted to another format for faster 3-D animation.
- Module for automatic generation of RRL programs using a CAD system. The user can

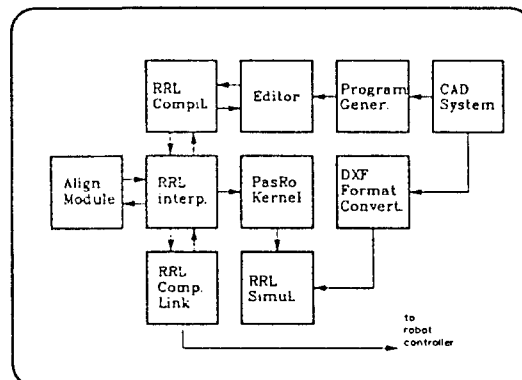


Figure 2: Structure and data flow of the RRL off-line system

define the robot trajectory in AutoCad on the selected layer. This module generates the RRL program for the defined trajectory.

3 Hardware implementation

3.1 RRL On-Line programming system

The RRL on-line programming system is implemented on the robot controller, based on a VME bus. The structure of the robot controller is presented in Fig 3. The organization of the controller hardware is hierarchical. The RRL system is implemented on the main CPU under O.S.-9. The main CPU features a Motorola 68020 with a Motorola 68881 arithmetical coprocessor for fast trajectory calculation and 1 Mb of RAM. The axis CPU is another Motorola 68010, where a digital servo controller with feed-forward compensation of velocity errors and PLC (programmable logic controller) controllers are implemented. All peripheral equipment such as D/A, A/D, incremental encoders, digital I/O are attached to the Motorola I/O bus. With such an organization, access to the peripherals equipment does not overload the VME bus, which is primarily intended for processor communication. Battery back-up RAM is used as an external memory device for saving application programs.

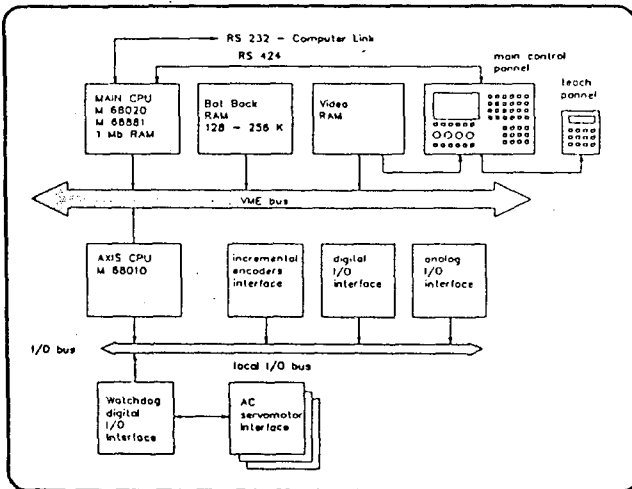


Figure 3: Block diagram of the robot controller

3.2 RRL Off-Line programming system

The RRL off-line programming system is implemented on a personal computer with MS-DOS O.S. and VAX or MicroVAX with VMS O.S. PC implementation offers wireframe models and wireframe representation without hidden line removal. In order to achieve reasonable simulation speed, PC implementation includes simplified dynamic models of the robot and actuators. PC implementation allows simulation of one 6. d.o.f robots, one object with 6. d.o.f and object with fixed coordinates during simulation. The simulation speed is determined by the complexity of the modelled environment. A typical configuration for PC includes a PC-AT 16 Mhz computer with arithmetical coprocessor, an EGA or VGA display and a display adapter. With such hardware, the simulation speed is about 3 times slower than execution of the same program on the real robot. VMS implementation is enhanced by interfacing to ROMAS - Robot Cell Modelling and Simulation System. The functions of ROMAS can be grouped into the following four modules:

- solid modeller for defining complex solids by translation, rotation and union of primitive solids;
- kinematic modeller for defining general kinematic chains having rotational, translational and parallelogram joints. Using this module we can define robots, grippers, positioners etc.
- cell modeller for definition of cell layouts using previously defined components and de-

scribing their spatial relationships and their connection types.

- robot cell simulation using a meta-language with commands for positional control of cell components, for defining changes in spatial and connection relationships, etc.

During simulation, ROMAS calculates the effects of program execution on the cell structure and generates the animated scene on the graphic terminal. When used with 3D graphics terminals (e.g. Tektronix 4236), it generates hidden lines or shaded display on-line. With its capabilities, ROMAS represents a general tool for choosing appropriate robots for specific tasks, evaluating alternative cell layouts and developing logically and positional correct program structures.

We have connected RRL and ROMAS to overcome the limitations of each system. Specifically, we can describe robot tasks in RRL using the target robot language and calculate the target robot dynamics. On the other hand, using ROMAS, we can quickly define complex cells, simulate simultaneous control of different robots and machines and evaluate different programs and cell layouts.

4 Example program

A sample RRL application program is shown in Figure 4. The task is to find the welding gap of the part and weld the part with weaving. For the sake of simplicity, the position of the welding gap is assumed to be unknown only in one direction and a tactile sensor is used to detect the edge, although the more general case can be easily programmed using frame arithmetics.

```
* sample program
maxsp = 1000 ; maximal speed [mm/s]
speed = 50 ; relative speed [%]
tcp 1 = 0 240 0 0 0 0 ; tool centre
set y = 20 ; variable for searching
* find edge of the part
appro to 1 for 0 -10 1
call find
frame 1 = frame 0 ; current robot frame
appro to 2 for 0 -10 1
call find
frame 2 = frame 0
* execute welding
```

```

appro to 1 for 0 5 5
smove to 1
speed = 2.5
wave 2 5 2 0
line to 2
depart for 0 5 5
speed = 50
home
stop
* subprogram for detecting the edge
* of the part with tactile sensor.
* Tactile sensor activates signal 4
label find
depart for 0 y 0 until sig 4 hi
if sig 4 lo then
    write text Cant'Find
stop
endif
return

```

Figure 4: Sample RRL program

Fig. 5 shows the graphic output of the simulation of the above program on a PC computer. Fig. 6 shows the graphic output of the simulation of the pick-and-place task with coordination with the CNC machine on a VAX computer with Tektronix 3236 graphics terminal.

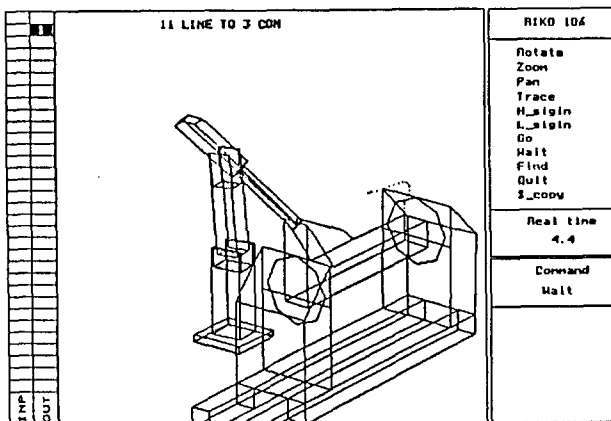


Figure 5: Examples of the graphic output of the simulation on the PC computer

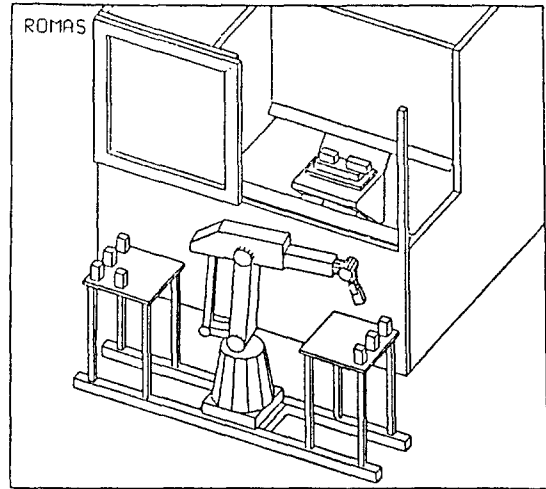


Figure 6: Example of the graphic output of the simulation on the VAX computer

5 Conclusion

RRL was demonstrated to be a successful, versatile and easy-to-learn robot programming system in many industrial applications. In addition, the RRL off-line module is an excellent tool for training and education. The RRL programming system was primary developed for the RIKO 106 electrical robot, intended for arc welding and assembly tasks. Due to the modular design of the system, it can be easily adapted for use in other robots. For a similar kinematic configuration, the system can be user-adapted by changing system parameters. For a completely different robot, only the kinematic transformation module has to be changed. Up to now, RRL was also installed on a 5-axis Cartesian robot for grinding plastic casts.

6 References

1. Bonner S., Shin K.G.: A Comparative Study of Robot Languages, IEEE Computer, Dec. 1982
2. Gruver W.A., Soroka B.I., Craig J.J, Turner T.T.: Industrial Robot Programming Languages: A Comparative Evaluation, IEEE Vol SMC-14, No4, 1984
3. Lozano-Perez T. : Robot Programming, IEEE Vol. 71. No. 7, 1982
4. Worn H., Stark G.: Robot Applications

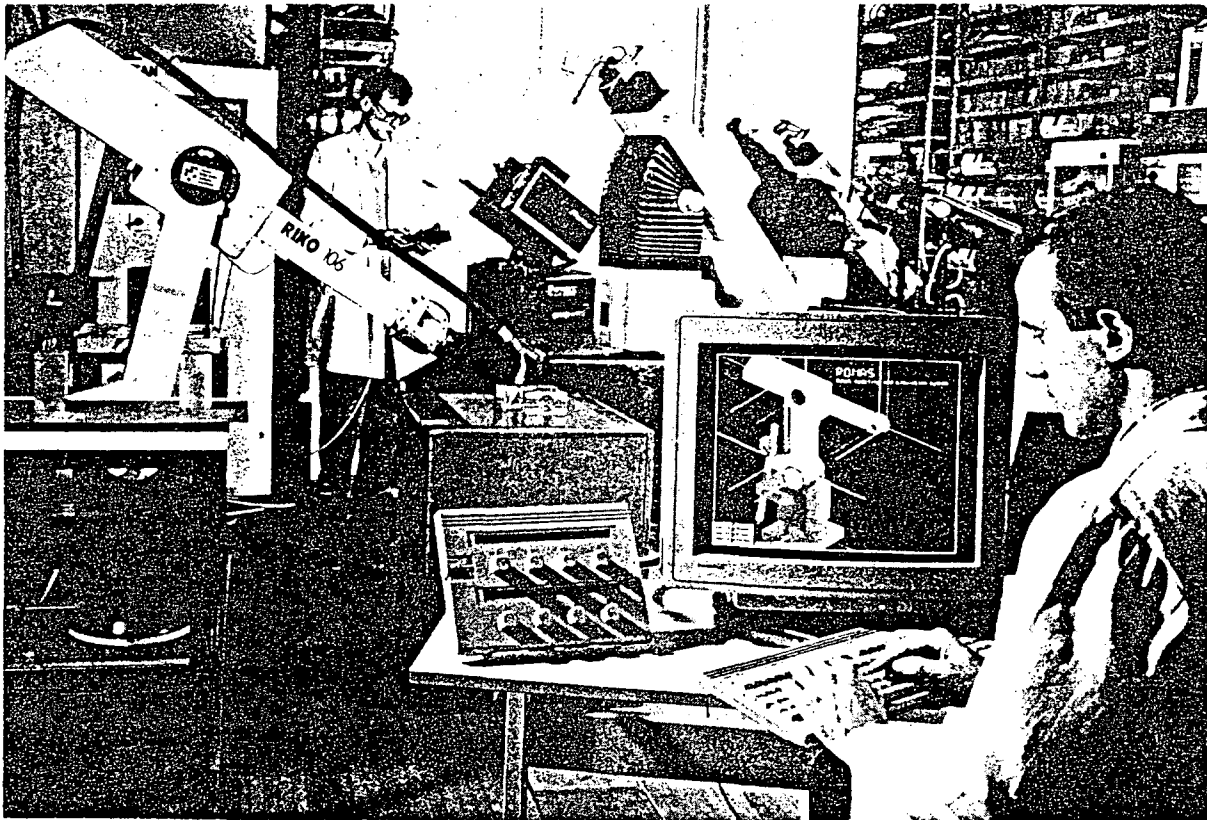


Figure 7: Riko 106 welding/assembly robot with robot controller and VAX based off-line programming system.

Supported by CAD Simulation, Robotics and Computer-Integrated Manufacturing, Vol3, No. 1, 1987

5. Milberg J., Schufer N., Tauber A.: Requirements for Advanced Robot Programming Systems, symp SyRoCo'88, Karlsruhe, 1988

6. Nemeč B., Lenarčič J.: A Robot Simulation System Based on Kinematic Analyses, Robotica, Vol. 3, 1985

7. Dombre E., Borrel P., Liegeois A.: A CAD

System for Programming and Simulation Robots' Action, Digital Systems for Industrial Automation, Vol. 2, No. 2, 1984

8. Sol E.J., van der Broek A. Th.M : Progress in CAD-Tools for Robot Based Flexible Automation Systems, 14th ISIR, Gotenburg, 1984

9. Blume C., Jakob W.: Programming Languages for Industrial Robots, Springer-Verlag, 1986

10 Paul R.P.: Robot Manipulators : Mathematics, Programming, and Control, MIT Press, Cambridge, 1981

Keywords: progress of computers, microelectronics, storage media, parallel computing, multimedia, personal computers

Matjaž Gams¹, Borut Hribovšek^{1,2}
¹Institut »Jožef Stefan«, Ljubljana
²ISKRA Elektrooptika, Ljubljana

ABSTRACT: In "Trends of Computer Progress" (Gams, Žitnik 1990) basic trends of computer progress were presented as an answer to growing speculations of either stagnation or spectacular breakthroughs. More detailed and technically supported survey is given in this second part, starting from microelectronics, storage media, parallel and distributed computing to operating systems and software technology, communications, multimedia, and finally PC trends. The overview strongly indicates that in the next 10 years progress will continue with similar astonishing pace as it did for the last 50 years (Baldi 1991).

POVZETEK: V članku "Trends of Computer Progress" (Gams, Žitnik 1990) so bile nakazane osnovne smeri razvoja računalnikov kot odgovor na naraščujoče špekulacije ali o stagnaciji ali o še bolj spektakularnem prodoru. V tem drugem delu so podane bolj natančne in tehnično podprte smernice razvoja mikroelektronike, pomnilniških medijev, paralelnega in porazdeljenega procesiranja, operacijskih sistemov, programske opreme, multimedijev in osebnih računalnikov. Pregled potrjuje zaključek, da se bo v naslednjih 10 letih razvoj nadaljeval z nezmanjšano hitrostjo (Baldi 1991).

1 Microelectronics

In the last 30 years ¹, since the industrial application took place, the progress of microelectronics is the essence of computer progress. An overview shows the following major indicators:

- Performance (density, speed) steadily increases while the cost per bit steadily decreases.
- Applications (penetration in new products) steadily increase.
- For the next 10 years, the mainstream technology is expected to be CMOS since it seems the best suited for VLSI or ULSI devices.

1.1 Performances

Integration density.

According to the Moore's law (Moore 1975), the

¹This survey is based on several magazines such as Byte, Future Generation Computer Systems, AI Magazine, etc.

number of transistors per chip has increased exponentially and has so far doubled every 1.5 years. The law can be observed from two lines in Figure 1 (source Intel) where the right line represents the progress in logical devices while the left one represents the progress in memory. The increase has been faster for memories due to more regular and simpler structure while logic devices require longer design times due to more complex functions.

Speed.

Gate speed constantly increases and already reaches 0.1 ns. However, while gate delay progressively decreases, circuit access time does not increase proportionally due to more and more complicated circuits.

Size.

Growing density and speed are based on the reduction in feature size and the increase of die size (Figure 2, source Intel). The growth was enabled by the ongoing progress in lithography and the improvements in the quality of materials.

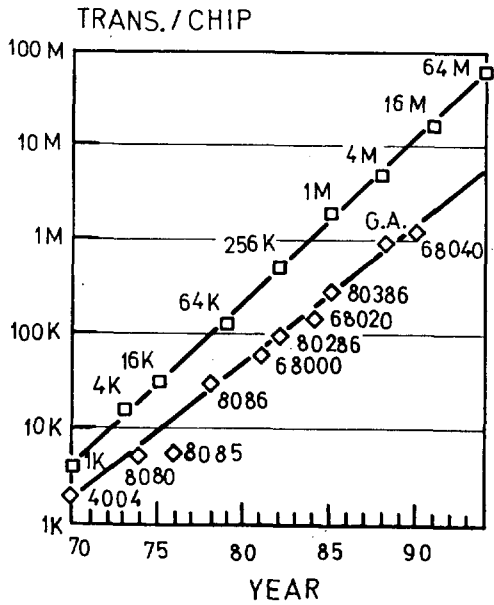


Figure 1: Plotting the number of transistors per chip versus time shows the Moore's law: every 1.5 years the integration density doubles.

1.2 Costs

In the last twenty years, the cost per bit exponentially decreases (Figure 3, source I.C.E. Report 1990). However, while the overall cost per bit decreases, process cost (i.e. production or equipment cost) actually increases and represents one of the biggest annoyances in future progress.

1.3 Applications

In recent years, microelectronics has become a key component of electronic equipment. Microelectron-

AREA (mm²)

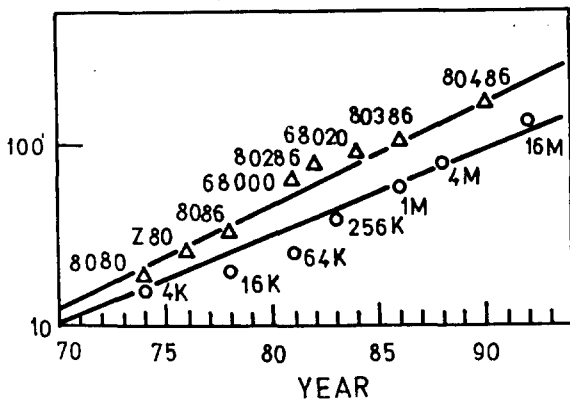


Figure 2: Die chip size versus time shows constant improvements.

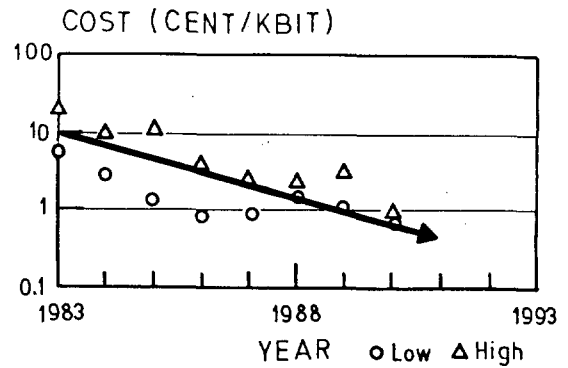


Figure 3: Cost per bit versus time for DRAM (Dynamic Random Access Memory).

ics have gained important new areas like video-recorders and CD's, while there is a constant growth in already established areas like computer applications. The ratio between semiconductors and equipment sales is presented in Figure 4 (I.C.E. Report 1990). The crossing point is near year 2150, when nearly all technical devices will be equipped by semiconductors. This broader use brings development of new features like analog circuitry or digitally controlled power capabilities and will effect every-day life in more or less every human activity, e.g. in smart houses.

1.4 Main technologies

Today, three main technologies are present on the market: bipolar, MOS (both based on silicon), and GaAs. The main technologies seem to be more complementary than competitive, each of them presenting a preferred field of application.

MOS (Metal-Oxide Semiconductor) technology consists of

- PMOS (Positive-well MOS) - obsolete,
- NMOS (Negative-well MOS) - being phased out,
- CMOS (Complementary MOS) - mainstream technology for VLSI, and

Semiconductors to Equipment Ratio Sales Trends

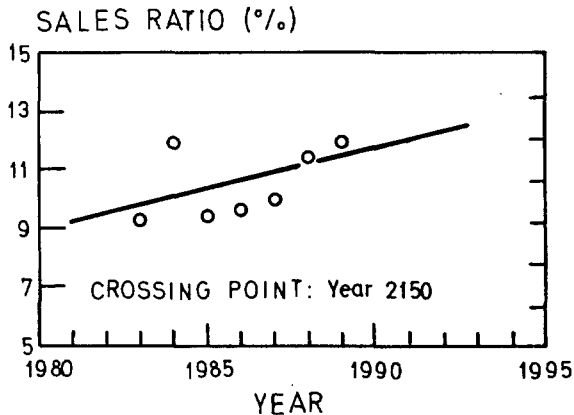


Figure 4: Ratio between semiconductors and equipment sales indicates an ongoing progress in semiconductor applications.

- BiCMOS (a hybrid of Bipolar and CMOS technologies) - in development.

CMOS devices are voltage-driven with relatively high threshold voltage, high input impedance and relatively low current drive capability. The main advantages are: low complexity, high input impedance and low power dissipation, which make them especially suited for large scale integration and modular circuit design. The drawbacks are low current driving capabilities, and relatively low speed.

Bipolar technology consists of

- ECL (Emitter Coupled Logic) - fastest silicon based process in growth,
- TTL (Transistor-Transistor Logic) - main bipolar logic technology phasing out, and
- LINEAR - mainstream analog technology in competition for complex devices.

Bipolar devices are current controlled devices with low threshold voltage, low input impedance and high current driving capabilities. The biggest advantages are: high current drive capability and good analog performances, which make them ideally suited for analog devices and for high speed logic. The most important drawbacks are high power dissipation, process complexity and low input impedance,

which do not allow modular design and the use for large scale integration.

GaAs devices

- OPTO - well defined market, expected to grow steadily,
- LOGIC - high costs confine it to very special applications

make use of several basic transistor structures. However, due to several problems the integration density is still by several orders of magnitude lower than for silicon based devices and the costs are much higher. On the other hand, the carrier mobility is more than 5 times faster than in silicon and the bandgap can be directly tailored. This makes GaAs ideally suited for optoelectronic and very high speed digital and analog applications, fields in which device cost is not a critical issue.

At present, MOS devices account for more than 50% of semiconductor's market (Figure 5, I.C.E. Report 1990) with GaAs covering only a meager 0.5%. In the next years, the total disappearance of PMOS and NMOS devices is expected. The share of CMOS and GaAs devices is further expected to increase.

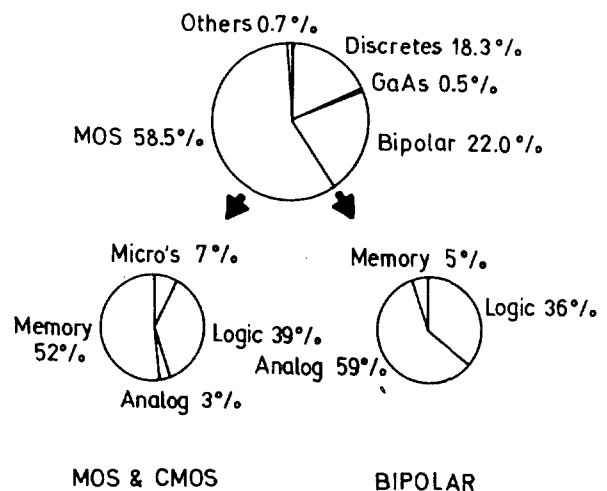


Figure 5: Split of market share among different technologies.

1.5 Present status and technological limits

Gate delays are in the 100-200 ps range for CMOS, and down to 25 ps for bipolar ECL technology.

While general extrapolation shows no major obstacle for further development, the following critical issues remain to be:

- lithography
The minimal feature size has been reduced from more than $10\ \mu\text{m}$ to less than $1\ \mu\text{m}$ in the last twenty years. The progress has been achieved through advances in optical lithography and, most importantly, no real barrier is likely to appear in the next ten years.
- transistor architecture
The first critical point has been reached for MOS transistors with device lengths around $1.2\ \mu\text{m}$ due to the increase in power density. While present solutions might go to $0.6 - 0.7\ \mu\text{m}$, new solutions can be expected at the level of the scaling rules and the reduction of the supply voltage around $3 - 3.5\ \text{V}$.
- interconnections
Since logic devices are becoming more and more complex, a certain limit of interconnections is becoming one of crucial factors. The solution seems to be the use of more metal levels, going from present two levels to three or more.
- defect density
In the last twenty years, the particle density in the production environment has been reduced by at least three orders of magnitude. While the production costs grow, the problem of defect density basically translates to the cost limitations.
- cost limitations
The cost of semiconductor facilities steadily grows and, perhaps, the darkest observation is that specific cost grow even faster. For example, the lithography costs versus feature size seem to be exponential.

1.6 The near future

Given the large amount of investments in the field, given the strong interconnections between microelectronics, the bulk of electronic industry, and the existing R&D prototypes, no revolutionary change will take place in the near future of around ten years. Performances, e.g., speed, capacity or complexity will continue to grow with exponential grow-

th as they did so far. Technology will continue to change as rapid or even faster than today. Therefore, due to constant introducing of new approaching and the growth in performances, microelectronic industry seems to be a young industry. However, from the point of investment, growing manufacturing costs, and a decrease of the number of semiconductor companies, it appears a mature industry branch with all its pluses and minuses.

2 Storage media

On today's market, devices with 4 Mbit DRAM's and 1Mbit SRAM's (Static RAM) can be obtained, and 16 Mbit DRAM's exist as prototypes. 64 Mbit DRAM's are being developed in development laboratories and basic elements for the future 256 Mbit chip are being studied in research laboratories. Following this projection and the one in Figure 1, it seems reasonable to assume that 256 Mbit DRAM's with a 0.25 micron geometry will be on the market by the end of the century. Therefore, personal computer memories will reach 100 Mbytes, workstation memories will reach 1 Gbyte, while mainframes will offer from 10 up to 100 Gbytes. In terms of logic, personal computers will reach 100 MIPS, workstations around 1GIPS and mainframes from 10 GIPS up to 100 GIPS. The difference between workstations and personal computers will be more in terms of price and purpose than in technological advances.

Besides microelectronics, several other areas like magnetic storage technology record important progress as well. Performances, such as density, access speed and transfer rates continue to improve. By the end of the century, head-disk interface should go beyond one tenth of a micron and aerial densities should grow from the current 100.000 bits per mm^2 to one megabit per mm^2 . With continued improvement in speed, capacity, and price/performance ratio, hard disk drives can and probably will still remain the preferred direct-access storage devices.

The biggest challenge to magnetic mass storage comes from optical technologies (Ryan 1990) such as CD-ROM, WORM (Write Once, Read Many times), and erasable optical disks. Optical storage is slower than magnetic primary because of the greater mass of optical read/write heads, while on the other hand it offers greater capacity. Quite

probably, optical storage will develop in parallel with magnetic storage and will be used in low-cost storage for low-end personal computers.

Therefore, the magnetic media hierarchy will basically remain unchanged in the years to come. The relationship between access and capacity is shown in Figure 6 (source Byte). The fastest technologies have the smallest capacity and the slowest technologies have the largest capacity. The pyramid makes a rough correlation between the height of each block and the percentage of each type of storage present in a typical system.

STORAGE HIERARCHY

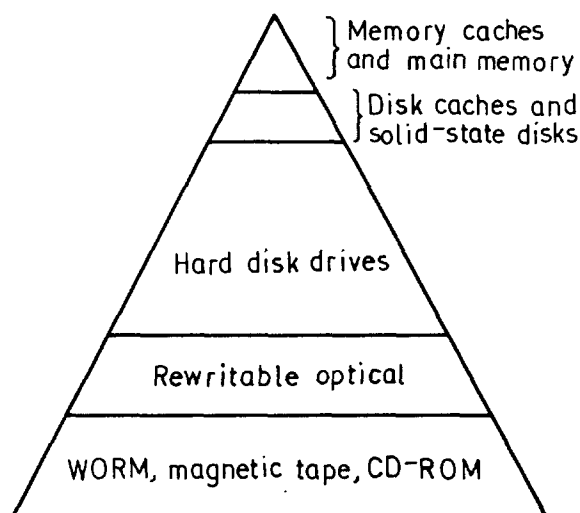


Figure 6: Storage hierarchy.

What impacts will these computer performances provide (Duby 1991)? Certainly, several tasks will be done in other way than today. For example, disk sorting techniques will be less important due to a simple fact that most of the sorting will be performed in the main memory. Simple calculation shows that 10 years from now even personal computers will have enough central memory to store (and sort) names and surnames of all people in Slovenia. Similarly, several hierarchical techniques will also be less often used. For example, relational databases will benefit from those memory sizes while hierarchical database management systems will be in decline. Large databases will influence every-day applications of artificial intelligence and knowledge-based systems.

3 Parallel and distributed computing

In next ten years, more and more attention will be devoted to parallel and distributed computing (Hertzberger 1991). At present, parallel computers with a large number of tightly coupled processors are commercially available and loosely coupled networks of computers are quite commonly used. However, classical Von Neumann computer architecture will probably remain dominant for at least 5-10 years since there are some difficult unsolved problems in parallel computing.

Some areas of parallel computing will evolve naturally with growing computer, microelectronic and transmission performances. For example, visualisation requires specialised processors and high speed communications requires quick accepting and storing large amounts of data. It is also quite likely that specialised artificial neural network processors will be used for pattern recognition tasks. Numerical intensive applications are another successful area for parallelism. Not surprisingly, since the parallelism in supercomputers, e.g., vectorization of applications, was an essential step to improve processing speed.

Different classes of parallelism are being identified:

- domain parallelism, where data structures are distributed among various processors,
- algorithmic parallelism, where the computer network is designed to match a particular algorithm and suitable code fragments and data flows are constructed, and
- task-parallelism, where a problem is divided into a large number of subproblems.

Until now, much less progress can be reported in general or even specialised symbolic application area. The largest experimental project where a large non-numeric application, i.e. a knowledge-base system, was the driving force behind the design of a parallel computer system, was the Japanese Fifth Generation Computer System. Results were mixed, with no breakthrough but with great impacts on future computer research and development. At present, several similar or competitive

projects run in Japan, Western Europe and USA. At first, logic programming language Prolog was utilised for implicit parallelism. In the second approach, programmers explicitly control parallelism usually by object oriented languages. What is used today is essentially an improvement of existing programming languages such as Modula or C++.

One of the fundamental problems is coordination and control of the communications among parallel fragments that comprise the task and one of the major bottlenecks in parallel computing is the lack of a coherent general model of describing and organising a parallel computing process. Lately, there were some promising attempts, unfortunately without greater practical meaning. In one approach (Valiant 1990), the possibility was shown of defining an idealised parallel processor PRAM, similar to the Turing's machine. It was also shown that under certain conditions, an algorithm runs n times faster on a parallel machine with n processors.

Therefore, at least theoretically it can be shown that parallel universality exists. On the other hand, practical solutions seem still quite far away as can be shown by a simple calculation: If only 1% of the whole process has to be solved in a sequential way, any parallel machine with any number of processors can not achieve an improvement by the factor 100.

Also, in the area of distributed computing some of the problems might slow down the expected progress, maybe simply by the unwillingness of many user communities to fully exploit the new possibilities by paying the price of forgetting the old-style approach and corresponding knowledge and skills. However, several implications seem inevitable, e.g., the role of supercomputers will quite probably decrease because of the cost-effectiveness of parallel and distributed computing.

4 Operating systems and software

In **operating systems**, less time will be devoted to memory allocation and more to new functions like distributed services and data, symbolic data query, cooperative processing and fault detection and recovery. After all, with 100 Mbytes on personal computers who needs virtual memory?

In the PC arena (Baran 1992), it is remarkable

how little operating systems have changed since the introduction of the IBM PC and, a couple of years later, the Macintosh. With the exception of the Mac OS, which has had an integrated GUI since its introduction in 1984, the big change in the operating-system arena in recent years has been the addition of windowing systems and GUI's to DOS and Unix, both of which traditionally have had command-line interfaces.

Of course, there is one obvious reason for the slow change in the operating-system technology, and this is compatibility with the huge data and applications base that already exists on millions of computers today.

Software technology is getting more complicated (Nance 1992). Developers have to hack through a jungle of computer languages, operating environments, user interfaces, and shifting standards to choose how to create their software. Therefore, one of important guides is toward standardisation and manageability of different aspects of software.

One of the main improvements will be more and more complex applications. Simulated models will replace experiments and enable efficient search for optimal solutions in many areas. Very large and complex data will enable access of all interesting data such as encyclopaedic data bases or business history for specific branches. Also, multimedia will be part of every-day activities combining mass products like television and personal computers. In industrial applications, CIM (CAD, CAP, CAM, CAQ and PPC) will become one of the most common and widely used approaches.

5 Communications and multimedia

Progress in communication technologies is expected to grow even faster than in computer technologies. In a decade, bandwidth possibility is expected to reach a few megabit per second from existing kilobits per second. It will effect architecture as well as disk needs since new transmission techniques will enable gigabytes per second access rates.

Due to the availability of many different kinds of data, standardisation will become pervasive in about every domain of computer technologies and applications from hardware to software, communi-

cations and user interfaces. Yet, communication with computers will become much more human-like and application-user-oriented since most of the users will basically have no knowledge about programming.

Multimedia uses the computer to integrate and control diverse electronic media (Robinson 1990) such as computer screens, CD-ROM disks, videodisk players, speech and audio synthesizers. Multimedia definitions run from combining text, sound, and animation onscreen to full digital video for editing and storage.

User interface will be one of areas where additional speed and capacities will enable great improvements. High definition graphic interface for both input and output will become pervasive while speech will enable high quality output as well as reasonable input. Since around 1 GIPS is needed for high speech and graphics performance, workstations near the end of the century will be able to do it at professional level while low-price personal computers will have to be contented with reasonable compromises. Commercial TV uses approximately 230 million bits per second and high quality personal computer displays need up to 500 million bits per second. This is far more than available disks on PC's enable: hard magnetic disks have about 16 million bits per second while CD optical disks enable 1.5 million bits per second. At the moment, CD's seem more prospective since they are removable and already a consumer product. The second problem is the storage capacity which at the moment enables only around one tenth of a minute of video data. However, there are two bright points: first is the expected growth in computer performances and the second improvement can be expected in compression techniques. If the loss of some fidelity is accepted (lossy compression) then even today it is possible to achieve compression rates of 50:1 for still images and up to 200:1 for moving video.

Therefore, expensive personal computers and workstations will achieve "marriage" with TV in forthcoming years. On the other hand, several new peripheral devices will become available such as smart scanners, readers and general data collectors from instruments and sensors. Most important, these new products will be task-user oriented for specific profiles. Also, color printing will considerably de-

velop. In display technologies, cathode ray tubes are expected to remain most widely used with evolutionary improved resolution.

6 PC world

Progress in personal computers is one of the most effective and, also, has strongly influenced all human activities. The first PC (Campbell 1990), as defined by IBM, processed approximately one-tenth of a MIPS (1 Million Instructions Per Second). In those days (the early 1980s), it costed about \$50.000 to put a theoretical 1 MIPS of processing power on your desktop. Today, 40 MIPS PSs are present on the market, and 100 MIPS PC's should arrive around 1995. This improvement in collective processing power will bring the cost per MIPS down below \$50 (Figure 7).

The nose-diving cost of raw computing power is the result of two factors: progress in microcomputer technology and the growing numbers of manufacturers of integrated system logic, graphics, I/O, and communications chip sets.

The early PC's were shipped with 4K-byte and then 16K-byte DRAM's. Today, PC's are routinely shipped with 8 Mbytes or more memory. Future machines will be designed to accept many megabytes of memory largely because today's application programs are starting to demand more memory space for data and for the programs themselves.

Also, significant trends in the PC world are: the drive for higher levels of integration, the sudden rise of alternative processors, multi-processor architecture and, also, several RISC-based (Reduced Instruction Set Computer) machines are making a play to become a factor in the PC world.

Despite enormous success, jumping up to the next level of personal computer performance may not happen smoothly. For example, the limit of around 50-100 MIPS seems quite a difficult one to overcome by existing PC technology. On the other hand, computers in general will tend to progress with similar speed as today and, quite probably, another technology will be introduced.

The explanation of lines in Figure 7 is as follows:

1. 8088/86 PC, 96 SSI chips, 4.77 to 8+MHz
2. 80286 PC, 4 VLSI, 40 SSI chips, 6 to 25 MHz
3. 80386 PC, 4 VLSI, 40 SSI chips, 16 to 25 MHz

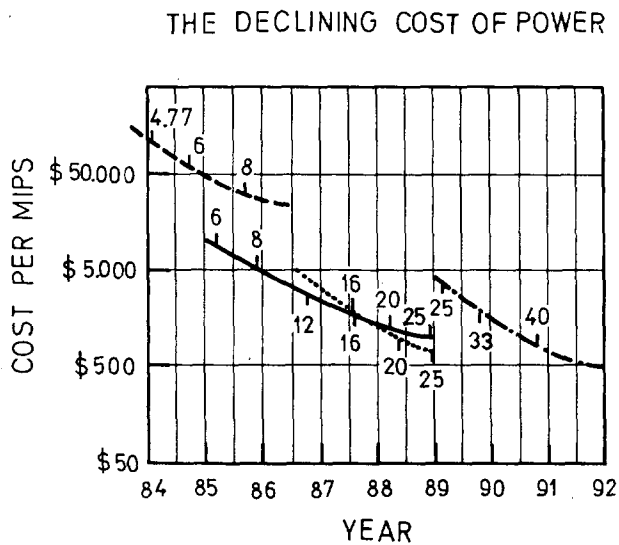


Figure 7: The declining cost of processing power.

4. 80386/486 PC (with cache), 3 VLSI, 19 SSI chips, 25 to 40+MHz

7 Summary

Exponential computer progress has been basically fuelled by the exponential growth in microelectronics. As a direct consequence, other computer-related activities progress with similar astonishing speed, among them storage media technology, parallel and distributed computing, software and operating systems, communications, multimedia, personal computers, databases, knowledge-based and artificial intelligence systems. This progress has been very constant over the last 50 years and was observed, for example, as the "Moore's law" in microelectronics. The scope of progress is expected to remain constant over the next 10 years and real technological limits are still far away.

Today, as a direct product of this astonishing development, a powerful PC on our desk has about 50 MIPS, 10 Mbytes of central memory and up to 1 Gbyte disk. Equipped with coprocessors and cache memory, it can compete with workstations designed a couple of years ago. Similarly, new powerful workstations approach performances of a couple of years old mainframe computers. For another comparison, today's PC's achieve performances of supercomputers designed 10 years ago. This means that

const-effectiveness improves faster for smaller machines and, consequently, the market share will continue to improve for PC's and workstations while the share of supercomputers and mainframe computers will continue to decline. Especially, when having in mind the expected progress in parallel, distributed computing and high speed, high rate transmission networks.

References

- Baldi L. (1991): Microelectronic trends, Future Generation Computer Systems 7.
- Baran N. (1992): Operating systems now and beyond, Byte, Special issue, January.
- Campbell G.A. (1990): Inventing the PC's future, Byte, Extra issue, January 20.
- Duby J.J. (1991): The evolution of information technologies in the 90s and its impact on applications, Future Generation Computer Systems 7.
- Gams M., Žitnik K. (1990): Trends of computer progress, Informatica 11.
- Hertzberger L.O. (1991): Trends in parallel and distributed computing, Future Generation Computer Systems 7.
- I.C.E. (1990): Mid-term 1990 status and forecast of the IC industry.
- Moore G.E. (1975): Progress in digital integrated electronics, Technical Digest of 1975 International Electronic Devices Meeting 11.
- Nance B. (1992): The future of software technology, Byte, Special issue, January 15.
- Robinson P. (1990): The four multimedia gospels, Byte, February.
- Ryan B. (1990): The once and future king, Byte, November.
- Valiant L.G. (1990): General purpose parallel architecture, Handbook of Theoretical Computer Science, ed. Leeuwen, North-Holland.