Flight Computer: Managing the Complexity

L.Bouchpan-Lerust-Juery
ESTEC/TEC-SWS:lionel.bouchpan-lerust-juery@esa.int
W.Gasti
ESTEC/TEC-ED:wahida.gasti@esa.int

Agenda

- Introduction
- Computer of Previous Generation
- Computer of Present Generation
- Future Trends
- Conclusion
- Questions ?

Introduction (1/3)

Academic Definition of a **microprocessor** (sometimes abbreviated **µP**)
is a digital component with transistors on a single semiconductor integrated circuit (**IC**).

IC → Moore's law formulation is the doubling of the number of transistors on ICs every 18 months
→ a rough measure of computer processing power

Digital Electronic were for decades evaluated through its integration capability MSI ⇨ LSI ⇨ VLSI

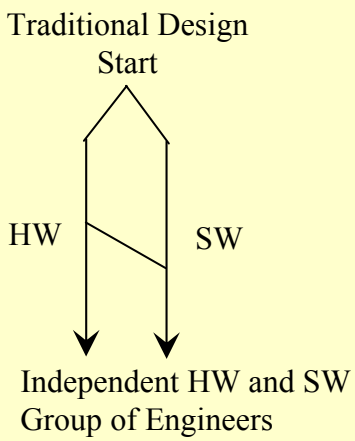
→ Advent of nanometer-scale integration and the increasing availability of transistors on a single die:
Production of more and more complex digital system

Today, **digital complex system** are no more evaluated at transistor level but at **functional level** (HW+ SW) or even rather at **system** level

System (Set of Function) - Implementation:

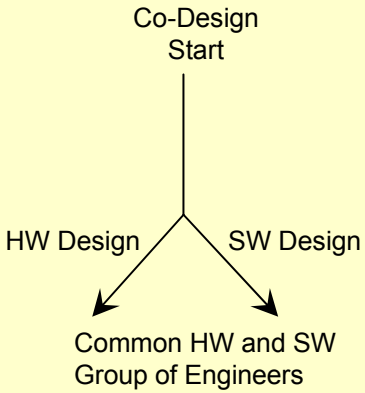
1. On ASIC ➔ System on Chip (**SoC**)
2. On FPGA ➔ System on Programmable Chip (**SoPC**)

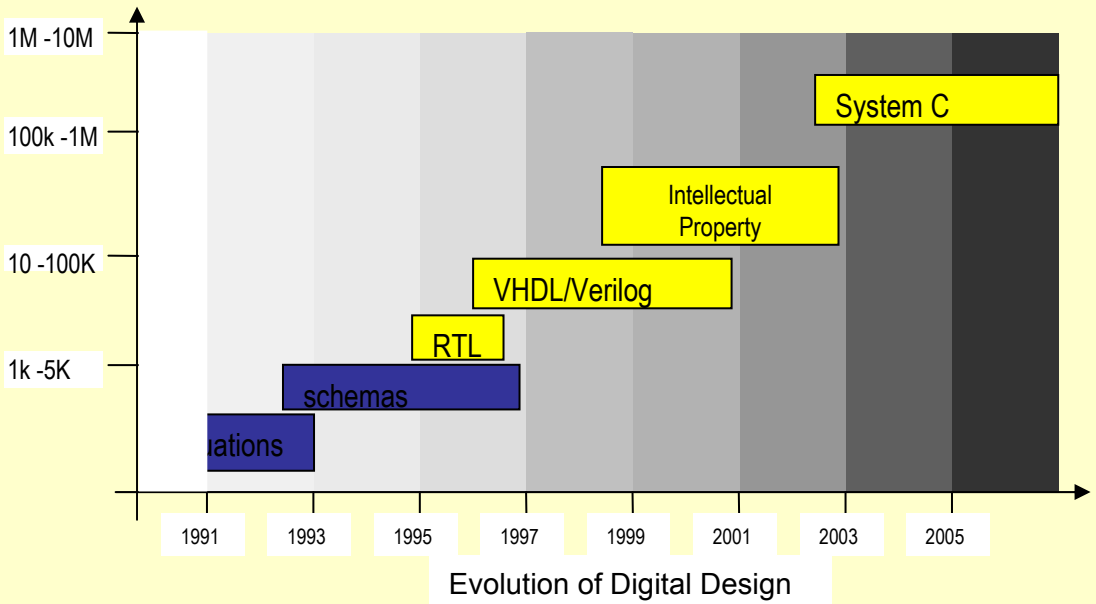
Conventional **Design Methodologies** that optimize HW efficiency (ASICs + COTS HW) do not necessarily optimize neither the **System** (HW + SW) development, nor the **quality and cost** of the final product



Introduction (2/3)

Researchers developed some basic approaches to the design of embedded software running on CPUs : **HW/SW Co-Design Methodology**.





HW/SW co-design Methodology Goals

- Unified Flow: Specification/Design/Implementation/Verification
- Hardware, Software, and Interface Synthesis
- Design re-use
- Reduction of TTM (Time To Market)

VHDL – Synthetiser: Designer concentrates on high level issues - Low level for Synthetiser

Free IP cores → http://www.opencores.org

C langage → System C (not yet mature)

Methodology

- Separation between function, and communication
- Unified refinable Formal Specification Model
  - facilitates system specification
  - implementation independent
- eases HW/SW trade-off evaluation and partitioning

Introduction (3/3)

Complexe Digital System → **Embedded Processor**

→ **Hardcore**
Performances to the detriment of Flexibility

→ **SoftCore** **Preference** goes to **Softcore** (avoid obsolescence)
Flexibility to the detriment of Performances

Open Softcore

- ✓ The Leon Processor
- ✓ The OpenRISC Processor
- ✓ The F-CPU Processor

Proprietary Softcore

- ✓ The Microblaze Processor of Xilinc
- ✓ The NIOS & NIOS II Processor of ALTERA

Others Softcore: clones of 6800, 68HCII, 68K, PIC:
http://www.opencores.org/browse.cgi/filter/category_microprocessor

Flight computers covers on board fault tolerant dependable computers.
Their main components:
- ✓ **µP**, I/O, bus, memory
- ✓ and related System SW & Application SW

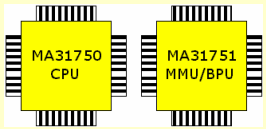
Agenda

- Introduction
- Computer of Previous Generation
- Computer of Present Generation
- Future Trends
- Conclusion
- Questions ?

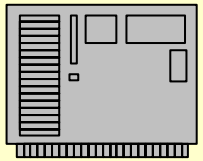
Past Generation Computer: MA31750

Hardware

1990: MA31750



Processor Module



Avionics Architecture

Software

- PSS05
- Doors
- HOOD + SDL
- Tight HW/SW integration
- No Operating System
- Assembly + Ada (TLD)
- SW size 100KBs

Missions

- Envisat,
- XMM,
- Rosetta, Mars-Express,Venus-Express,
- MetOP
- …etc

Agenda

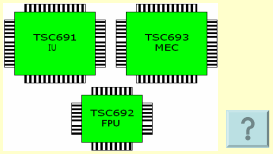
- Introduction
- Computer of Previous Generation
- Computer of Present Generation
- Future Trends
- Conclusion
- Questions ?

Present Generation Computer: SPARC Family

Hardware

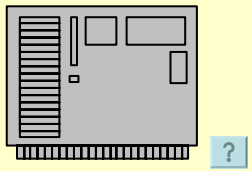
SPARC 7 Architecture

1995: 3-Chip-Set: ERC32

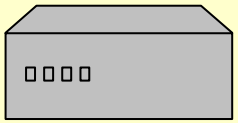


2000: ERC32 single chip:TSC695

Processor Module



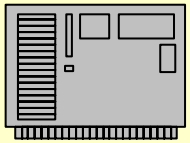
Avionics Architecture



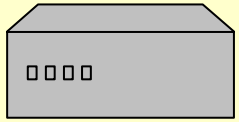
SPARC 8 Architecture

2005: AT697

Processor Module



Avionics Architecture



Software for SPARC 7

- ECSS E40
- Doors
- UML 1.x
- Loose HW/SW integration
- Operating System (RTEMS)
- Assembly + C (Heavy use of Open source tools)
- SW size in the 1MBs
- Many companies having different level of expertise
- Shoehorning the SW into the computer
- ISVV intensive

Missions

- Cryosat,
- Goce,
- ADM-Aeolus,
- Herschell-Planck
- …etc

Agenda

- Introduction
- Computer of Previous Generation
- Computer of Present Generation
- Future Trends
- Conclusion
- Questions ?

Future Trends (1/5)

Next Generation µP

for Flight Computer

- ASSP based on SPARCx or Advanced CPU core
- SoPC — Cost, FPGA PA issues
- SoC
  - Single core (which core ?)
  - Multi-core (Homogeneous cores MPSoC ?)
  - Heterogeneous cores (Network on Chip NoC ?)

Key Features for Flight **µP** Design

- Radiation Tolerant Processor
- Microelectronics –IP library
  - SEU/SET mitigation techniques
- Buses/Networks

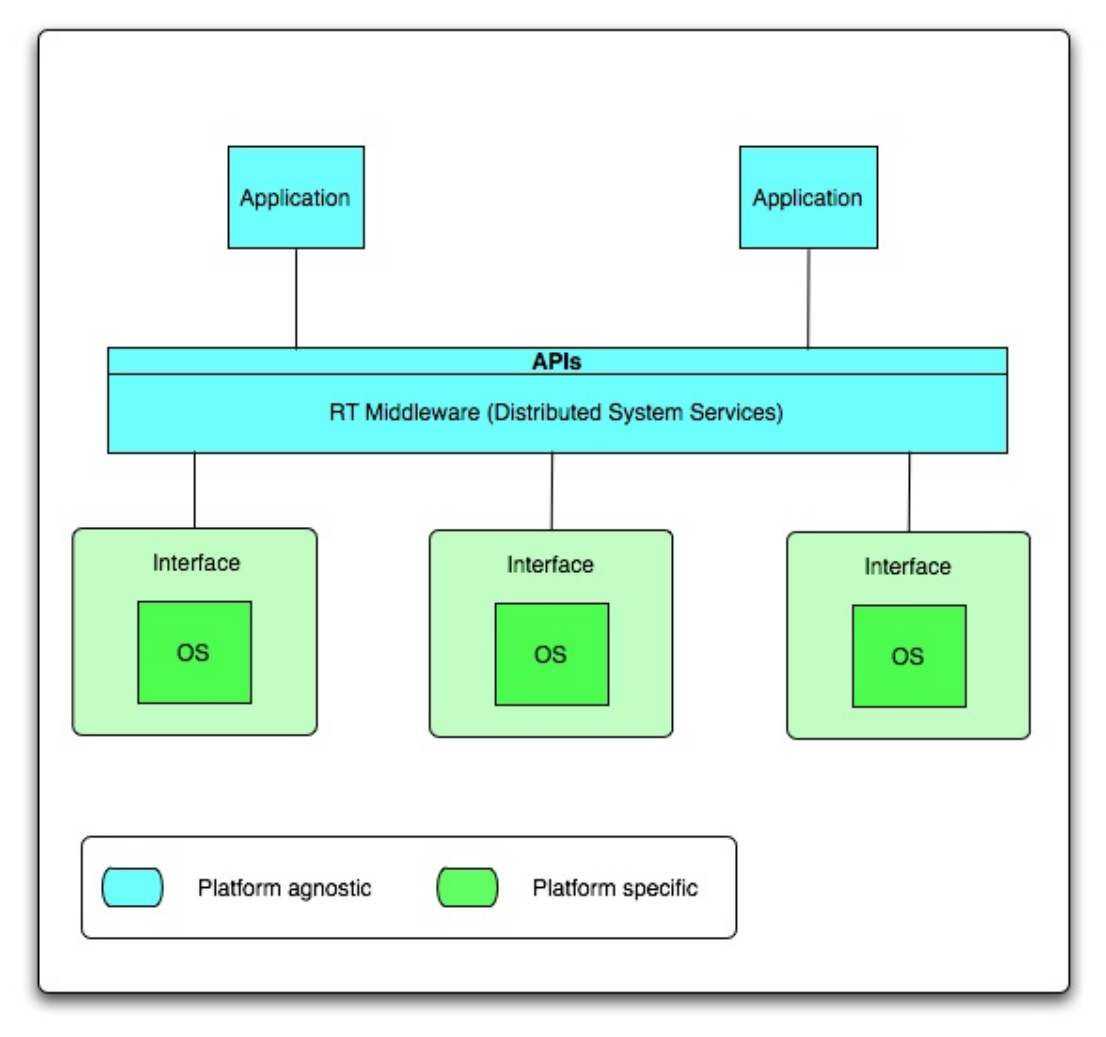
Designers of Future On board computers (**On-Board System**) would have to handle large pre-designed CPUs, and SW would have to be treated as a first-class component in the system design.

Future Trends (2/5)

- E40 very likely
- Even more companies in the SW with different level of expertise
- Transition from UML1.x to UML2.0 (SysML + Space specific profiles)
- SW size in the 100 MB.
- Reuse of existing building blocks
- Huge ISVV effort
- Need tools to help the SW engineer masterise the complexities of interactions and layers
- Automated tools to help validate the diverse parts of the SW wrt the target platform => HW / SW codesign almost mandatory => SW at System level using the industry's best practice.
- Integration of the HW and SW tools
- With such a SW complexity, not possible anymore to reverse engineer the Reqs. from the code => Literate programming is an approach
- The SW developer does not run the show anymore. SW is no more an artisanat but an industrial product.

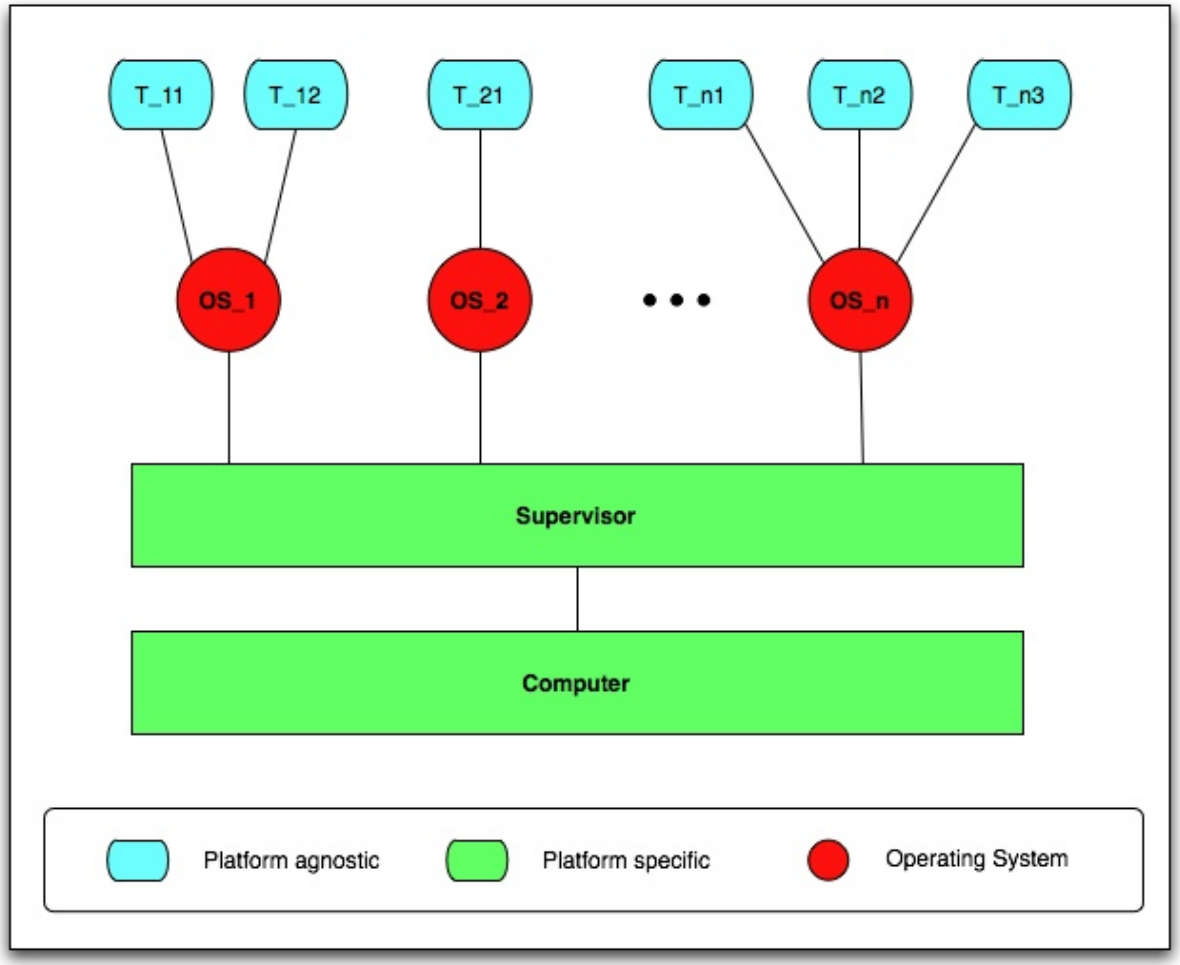
Future Trends (3/5)

Middleware (Distributed SW among CPU and CPU cores, will very likely bring the use of a Middleware)



Future Trends (4/5)

Virtualization (Partitioning and virtualization (mono or multi OSes))



Future Trends (5/5)

Literate Programming

7. Several variables are needed to govern the output process. When we begin to print a new page, the variable *page_number* will be the ordinal number of that page, and *page_offset* will be such that $p[page\_offset]$ is the first prime to be printed. Similarly, $p[row\_offset]$ will be the first prime in a given row.

[Notice the notation '+ ≡' below; this indicates that the present section has the same name as a previous section, so the program text will be appended to some text that was previously specified.]

```
⟨ Variables of the program 4 ⟩ +≡
page_number: integer;   { one more than the number
        of pages printed so far }
page_offset: integer;   { index into p for the first entry
        on the current page }
row_offset: integer;    { index into p for the first entry
        in the current row }
c: 0 .. cc;   { runs through the columns in a row }
```

8. Now that appropriate auxiliary variables have been introduced, the process of outputting table p almost writes itself.

10. The first row will contain

$$p[1],\ p[1+rr],\ p[1+2*rr],\ \ldots;$$

a similar pattern holds for each value of the *row_offset*.

```
⟨ Output a line of answers 10 ⟩ ≡
  begin for c ← 0 to cc − 1 do
    if row_offset + c * rr ≤ m then
      print_entry(p[row_offset + c * rr]);
  new_line;
  end
```

This code is used in section 9.

22. The inner loop. Our remaining task is to determine whether or not a given integer j is prime. The general outline of this part of the program is quite simple, using the value of *ord* as described above.

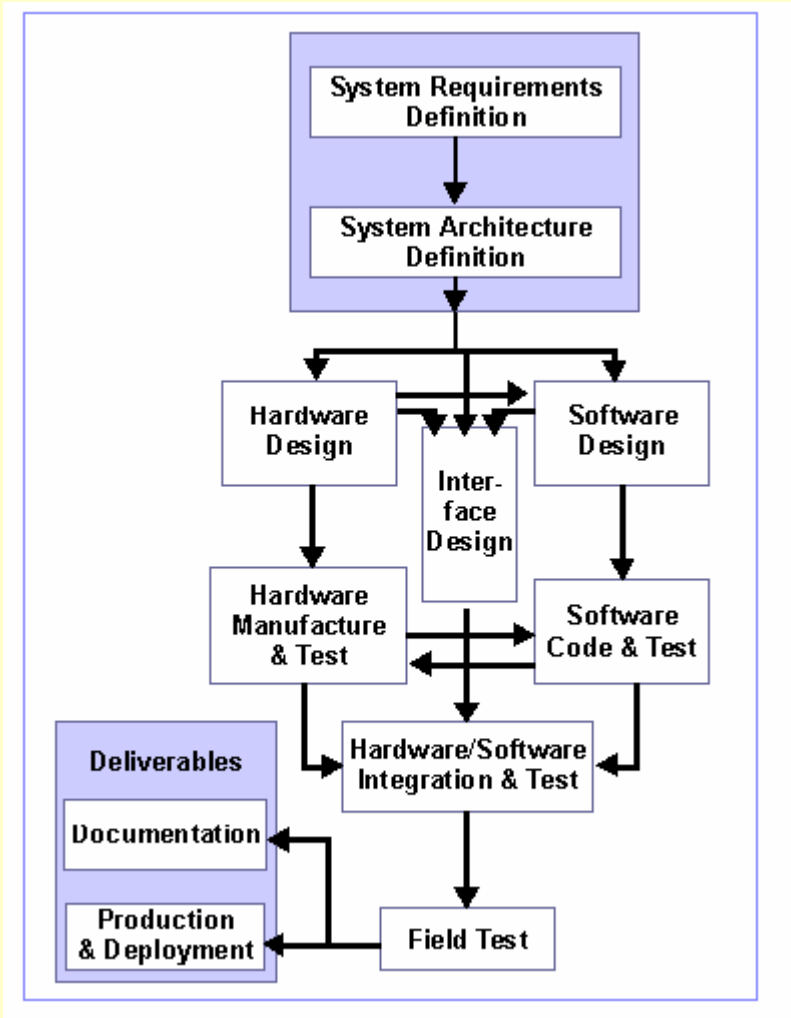
```
⟨ Give to j_prime the meaning: j is a prime
        number 22 ⟩ ≡
  n ← 2; j_prime ← true;
  while (n < ord) ∧ j_prime do
    begin ⟨ If p[n] is a factor of j, set
          j_prime ← false 26 ⟩;
    n ← n + 1;
    end
```

This code is used in section 14.

23. ⟨ Variables of the program 4 ⟩ +≡

```
n: 2 .. ord_max;
        { runs from 2 to ord when testing divisibility }
```

New Paradigm for Flight Computer



Next Generation µ-Processor (s)

- ✓ **Solution is Cost/Schedule-Driven**
- ✓ **Design Methodology should be well-thought-out**

Is it a Dream or Reality for Flight Computer ?

Next Generation Flight Computer

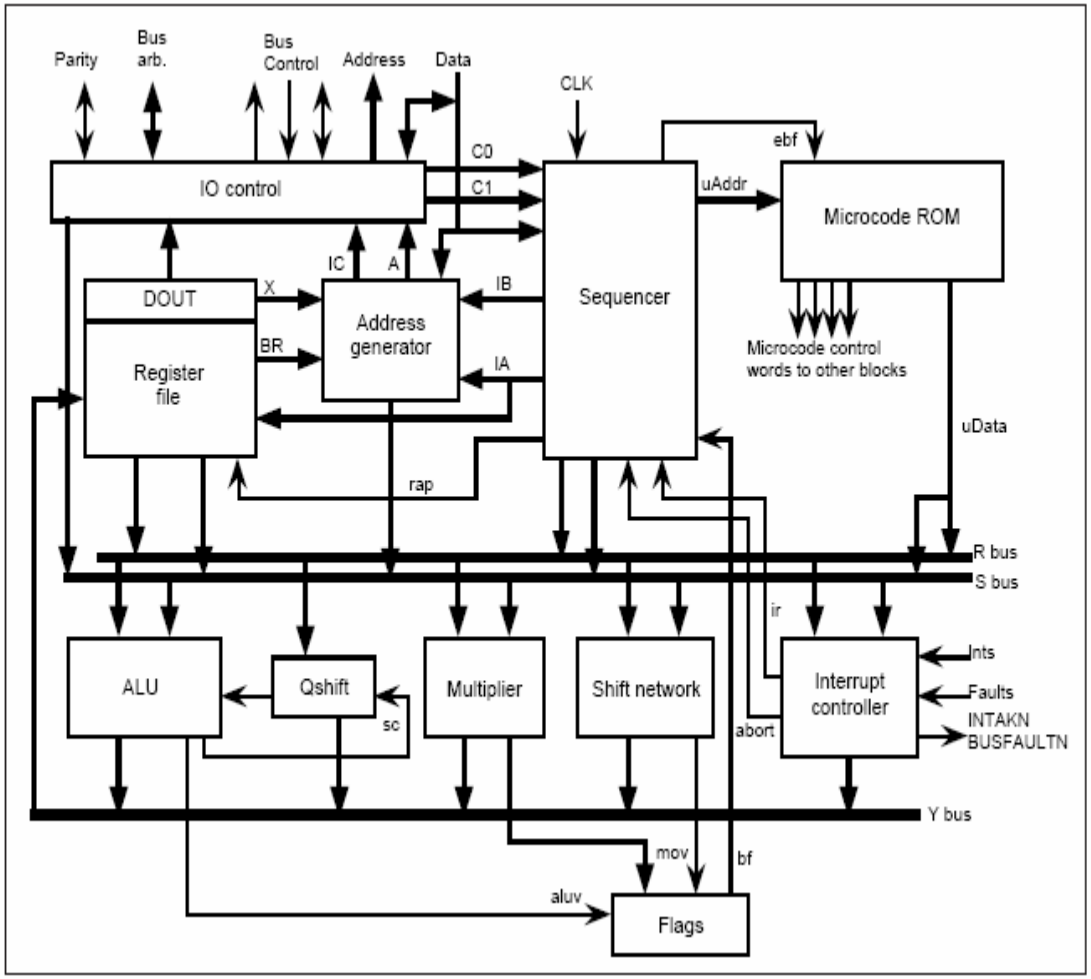
Managing the increasing Complexity

Out of the Findings of This Round Table

THAT'S ALL FOLKS !

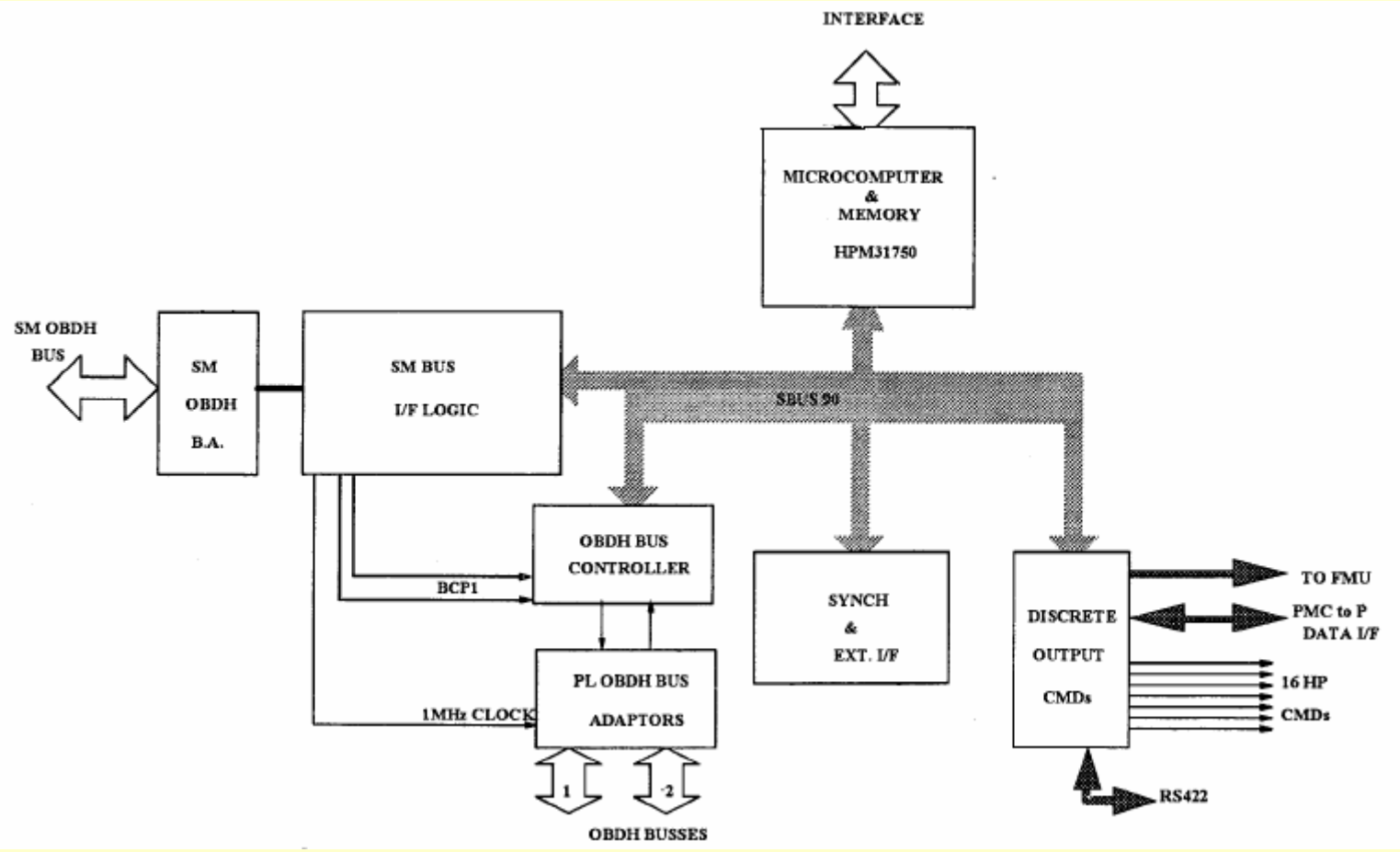
QUESTIONS ?

MA31750 Internal Architecture

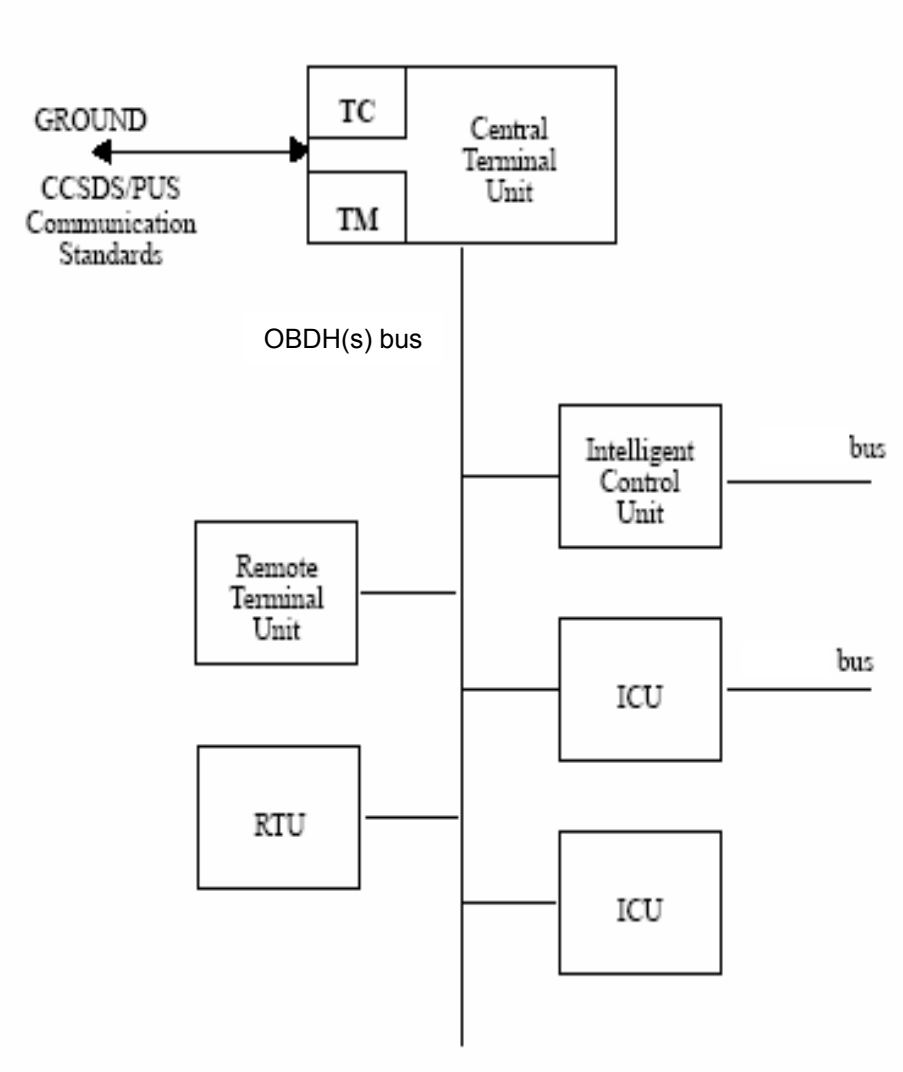


- CSIC (Mil1750 Inst. set)
- Two chips
- 84 and 68 pins
- 16 bit
- 2 MIPS at 16 MHz
- CMOS/SOS 1.25 µ
- Designed and manufactured by GPS
- Full European design
- Largely sold and used in Europe, in USA and other countries
- Still under production at DYNEX Semiconductor

MA31750 Processor Module



Previous Generation Computer



ERC32 3-Chip Set Architecture

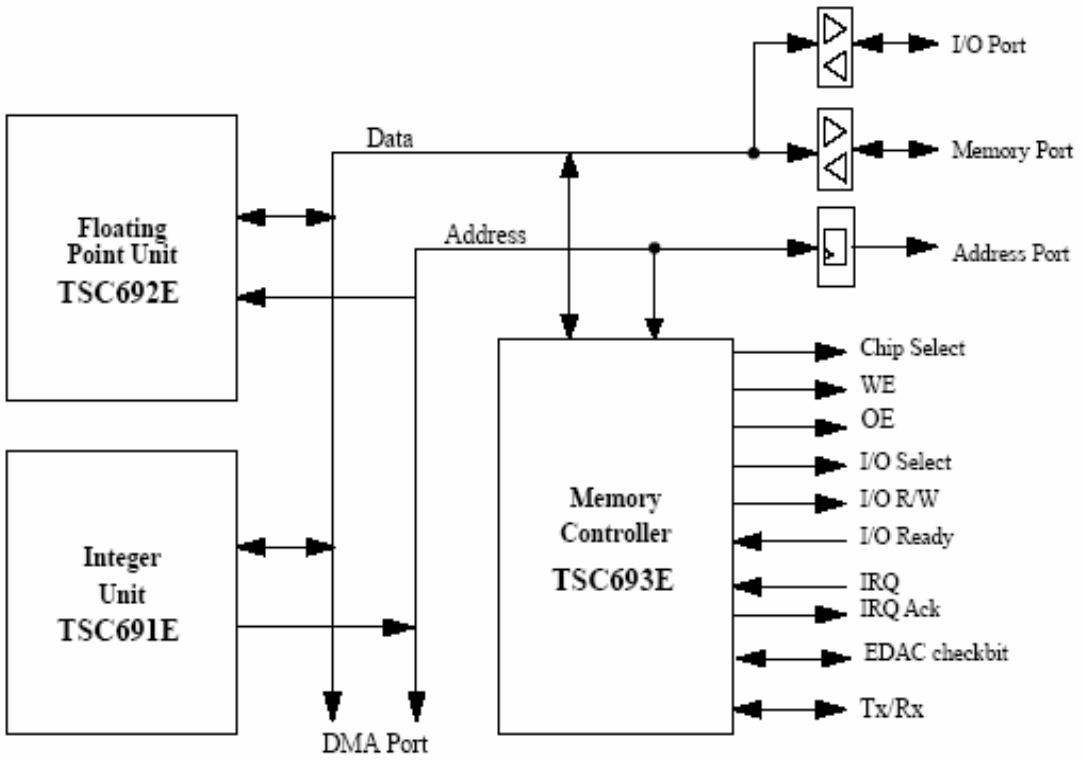
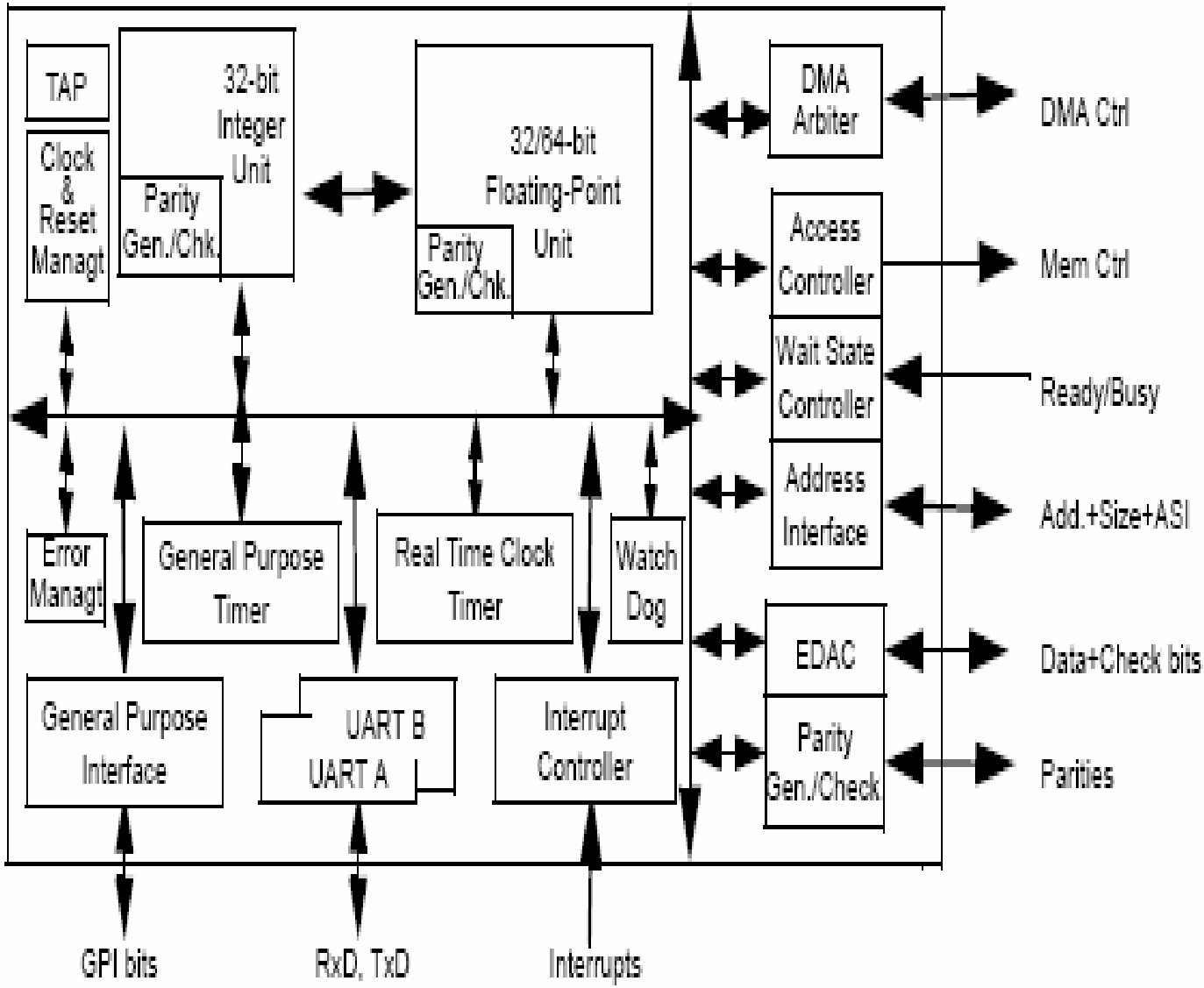


Figure 1. ERC32 Architecture

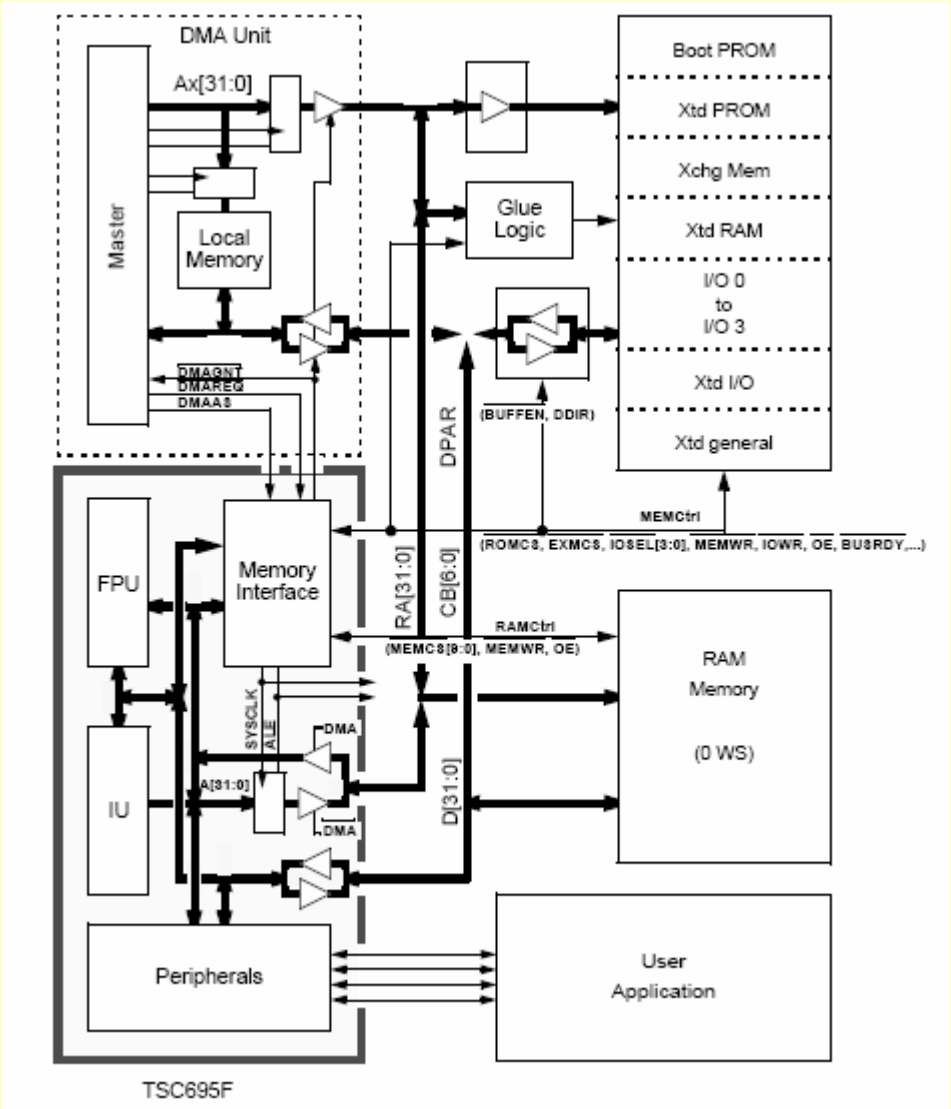
- SPARC V7, 32 bit
- RISC
- Three chips
- 256, 160, and 256 pins
- 10 MIPS at 14 MHz
- Manufactured by TEMIC
- CMOS RT 0.8 micron
- Partial European design
- Used in man space systems:DMS-R, SPLC, ERA, ATV,and also in PROBA
- Phase-out: Last buy June 2002

TSC695 Internal Architecture

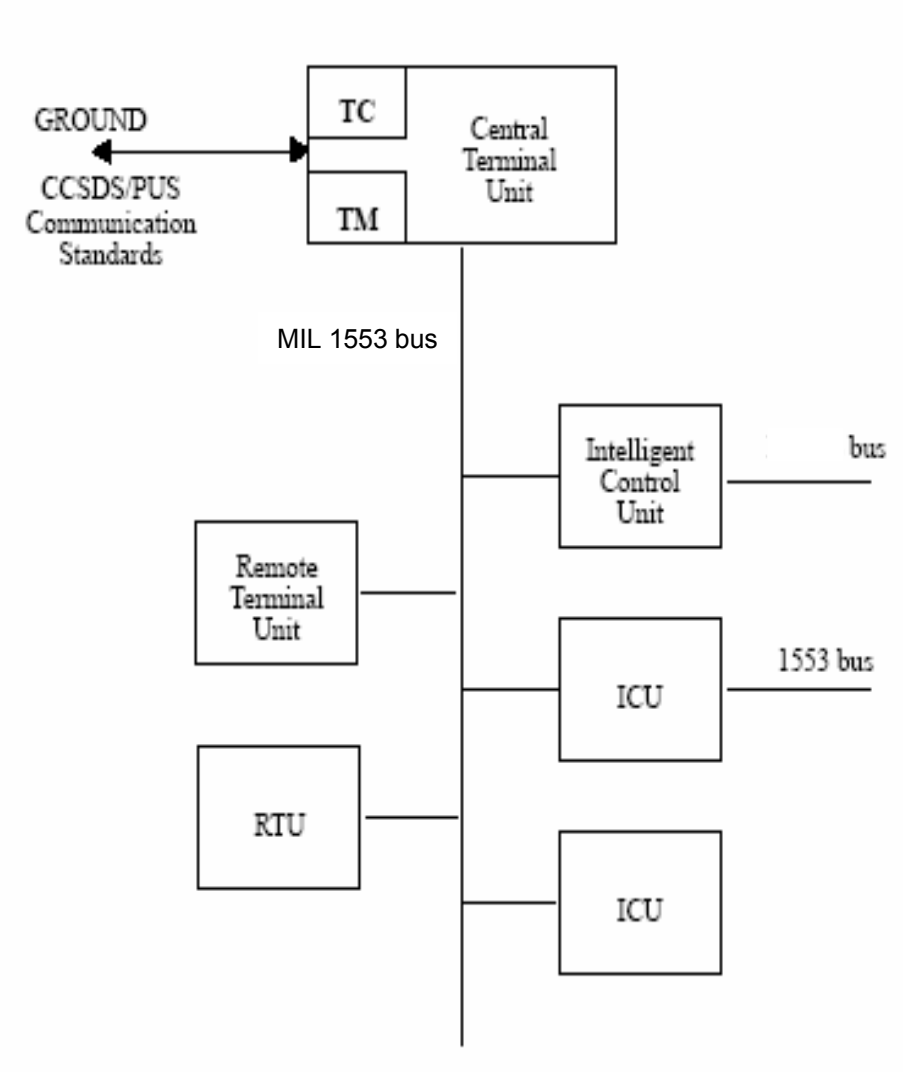


- SPARC V7, 32 bit
- RISC
- Single chips of 256 pins
- 20 MIPS at 25 MHz
- Manufactured by TEMIC
- SCMOS RT Plus 0.5 micron
- Merging of the ERC32 3-chip set
- Removing of unused functions
- Addition of new functions and bug fixes

TSC695 Processor Module

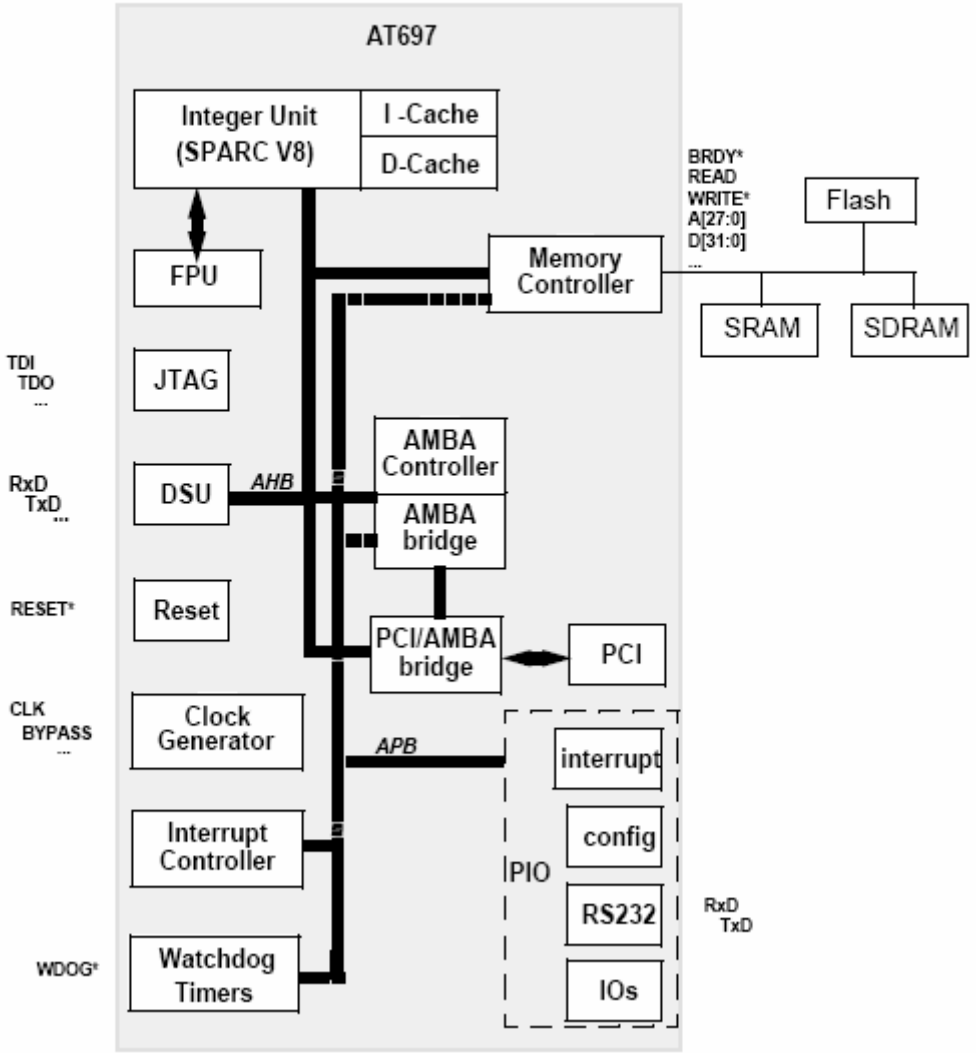


Present Generation Computer



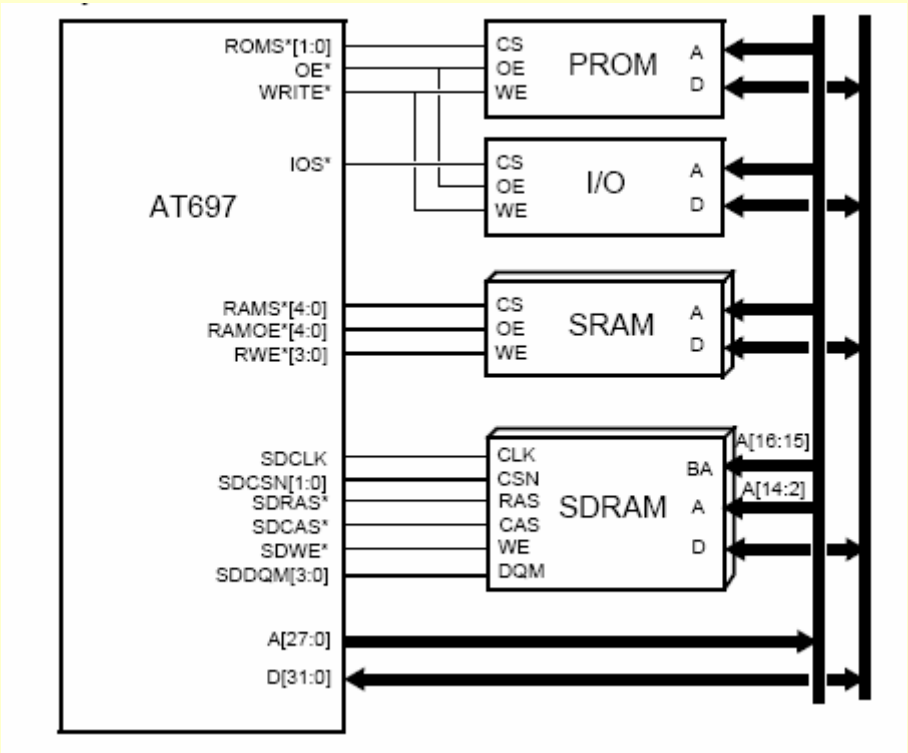
AT697 Internal Architecture

Figure 1. AT697 Block Diagram

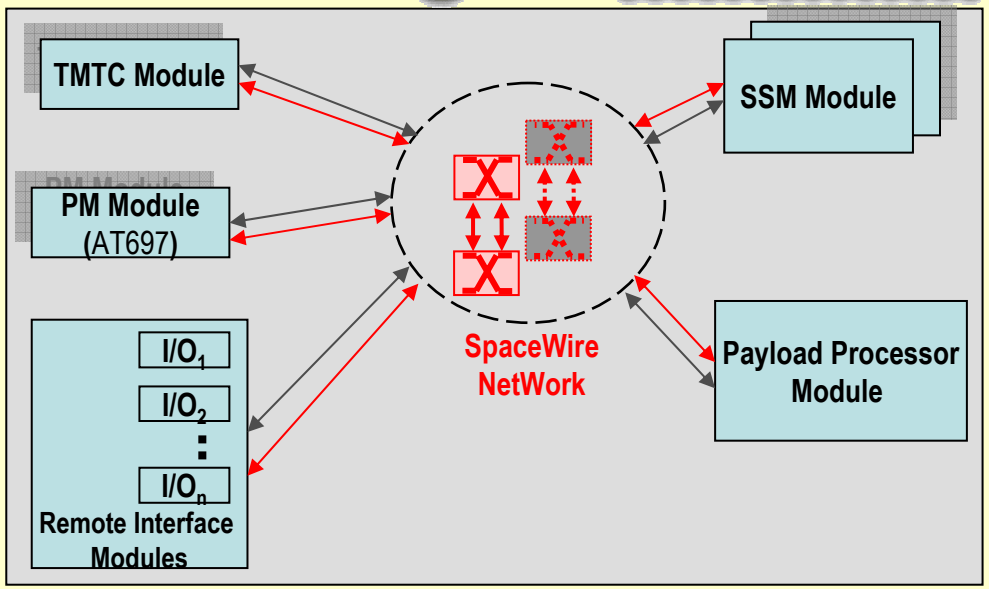


- SPARC V8, 32 bit
- RISC•Single Chip
- 349 pins package CLGA
- 100Mhz, 100 MIPS
- Use of caches:
  16 Kbytes Data cache, 32 Kbytes Inst cache
- LEON core designed by ESA
- Manufactured by ATMEL
- CMOS: AT58KRHA 0.18 micron
- Use of third party IP cores (PCI, FPU)

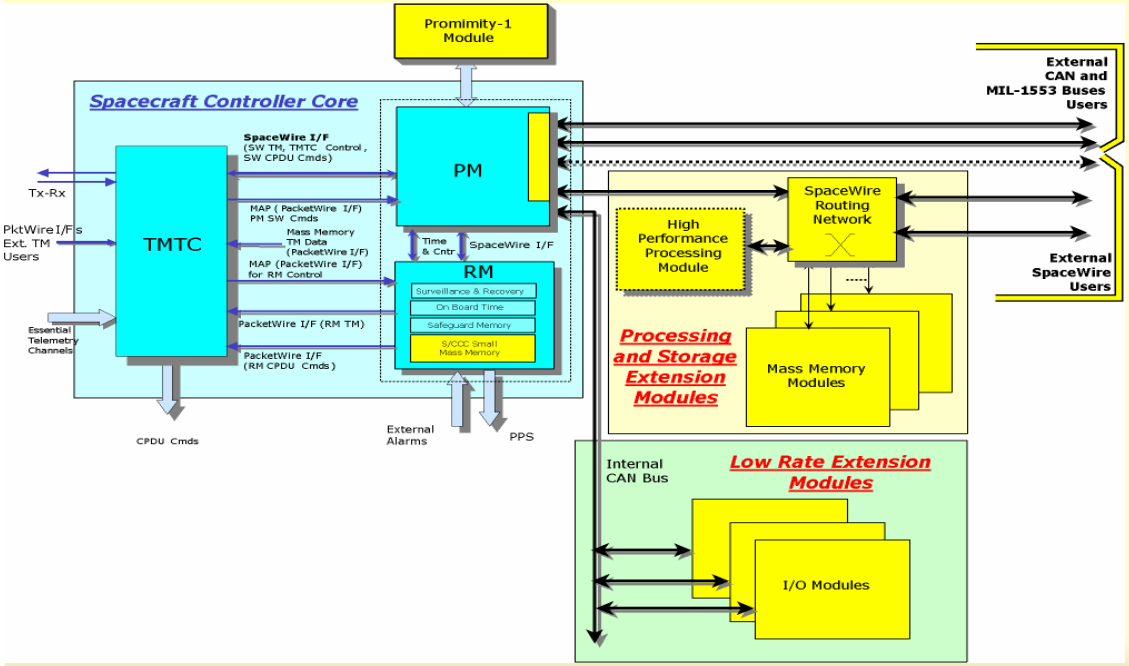
AT697 Processor Module



Present Generation Computer



Compact Data Handling System



Distributed Data Handling System