Inputs to new processor requirements

Saab Space **Austrian Aerospace**

Applications analysed

- **Spacecraft Management**
  - Data Handling
  - Attitude control and propulsion
  - Platform management (power, thermal)
  - Time keeping using GPS/Galileo signals
- **Payload management**
  - Instrument control
  - Science data storage and downlinking
- **Instrument processing**
- **Launcher control**

Performance requirements

- **Raw processing performance driven by payload processing but difficult to state firm numbers**
  - Payload processing often uses what is available
  - Architecture flexibility often better than raw speed
  - Multiple CPUs or application specific co/pre-processors sometimes better than a single CPU due to the nature of the problem
- **Platform computing more driven by deterministic behaviour**
  - Important to minimise interaction between different applications
- **Deriving future requirements by extrapolation is probably the method giving most realistic goals**

- **Conclusion:**
  - Do the best possible within the available budget!
  - Consider easily achieved DSP needs during selection and architecture trade-offs

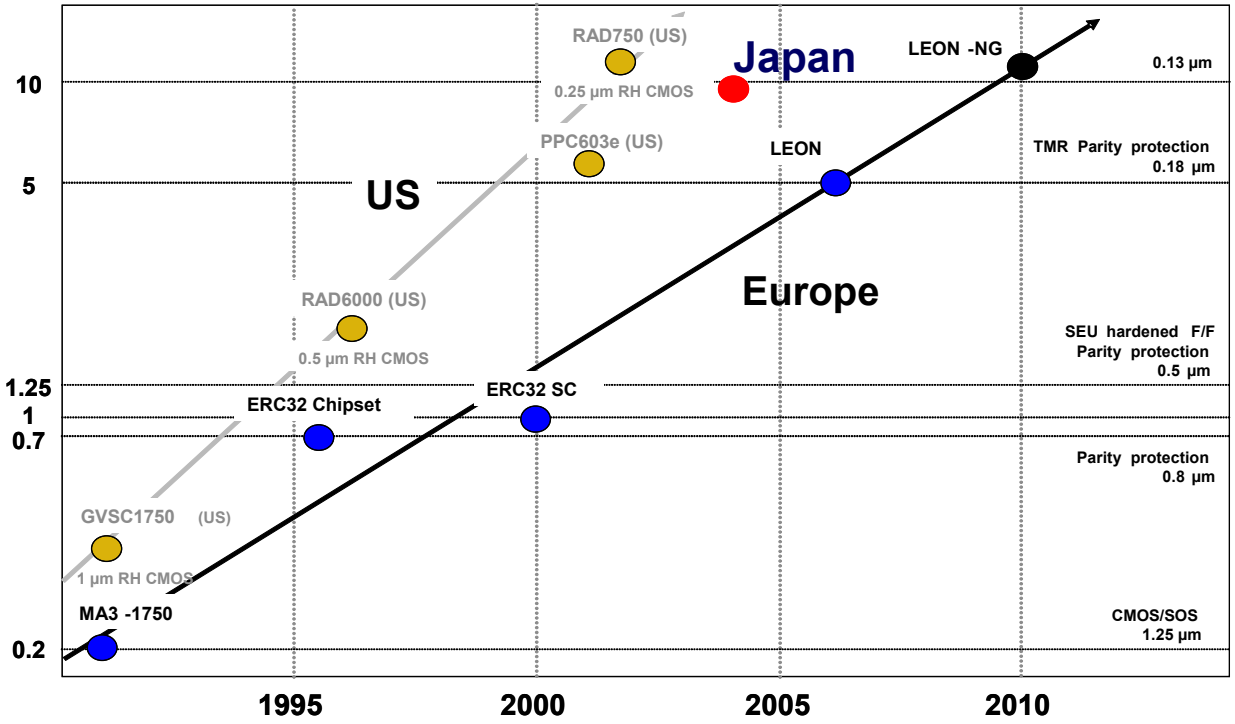
CPU architecture considerations

- Multi-core CPU needed for high computing power when parallel processing is possible or to split different applications on different cores
- In case a common main memory is used for the different cores a Memory Management function is mandatory to prevent different applications from interfering with each other in case of errors.
- On-chip memory or large data caches with write-back capability allows for high performance and reduced interaction also in computing-intensive applications with a shared memory. Cache sizes as large as practically possible but not less than 2xAT697
- On-chip RAM-based FPGA desirable to allow for addition of coprocessors in payload applications
- Powerful cache control mechanisms like locking needed to improve predictability (significant improvement of the AT697 functions needed)
- The selected architecture should not preclude typical DSP application

CPU architecture considerations

New Europe goal?

- Performance by extrapolation:
- Are European foundries competitive?
- Does the SPARC architecture give sufficient performance?



- Eurospace Oct 05

- Anything else we can do better in Europe?
- Do we have the resources to participate in the performance race?
- What will be the consequences if we do not participate?

CPU architecture considerations

- **If a SPARC architecture is selected, do we need the V8E extension for embedded real-time systems to improve real-time performance?**

- **If a SPARC architecture is selected, are more register windows required?**

- **No backwards compatibility really needed for the CPU part as long as compilers, debug tools and simulators are available.**

  It is mainly important to be able to keep existing S/W interface to the application specific I/O system (compare upgrading from MA31750 to ERC32 to AT697, the differences in instruction set and on-chip CPU registers are fairly easy to handle, differences in I/O system much more costly.

Software considerations

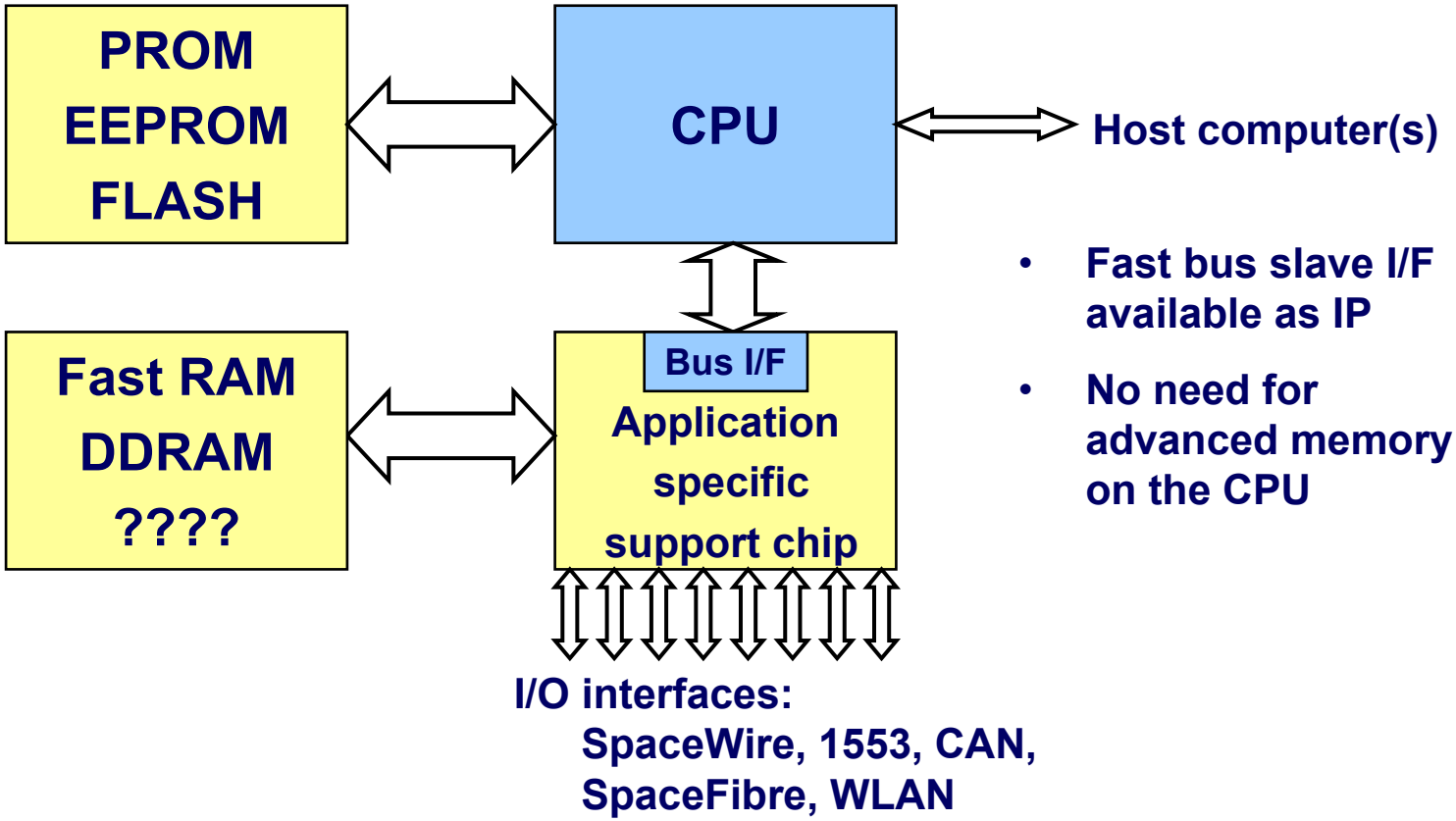
- Observability is important, the AT697 DSU function is a first step but must be significantly improved
- Timing and schedulability analysis tools and methods must be available and approved by ESA <u>before</u> any project starts
- Necessary to find an OS with multiprocessor and MMU support
- Have the compiler and OS qualified for space use before any projects start!
- Simulator to be developed in parallel with the hardware. To be validated and approved by ESA.

Hardware considerations

- **Small package desirable, should be kept about a 472 CLGA**
- **Interfaces and on-chip peripherals:**
  - High-speed (>10 Gbps) bus for fast memory and I/O accesses (parallel bus similar to Pentium front-side bus, PCI Express, …)
  - Low-speed (< 600 Mbps) simple parallel bus for “slow” memories and ”slow” I/O
  - Fast debug link allowing real-time dump of DSU buffer for each CPU - UART for legacy compatibility
  - Simple watchdog
  - RTC timer synchronisable to an external reference (e.g. PPS signal)
- **Power consumption scalable from a few 100 mW to 4W**
- **Other considerations:**
  - Serial EPROM I/F ?
  - No advanced memory I/F
  (commercial technology advances too fast making the CPU obsolete very quickly)
  - Local memory/large caches more important than lots of I/O

Possible hardware environments

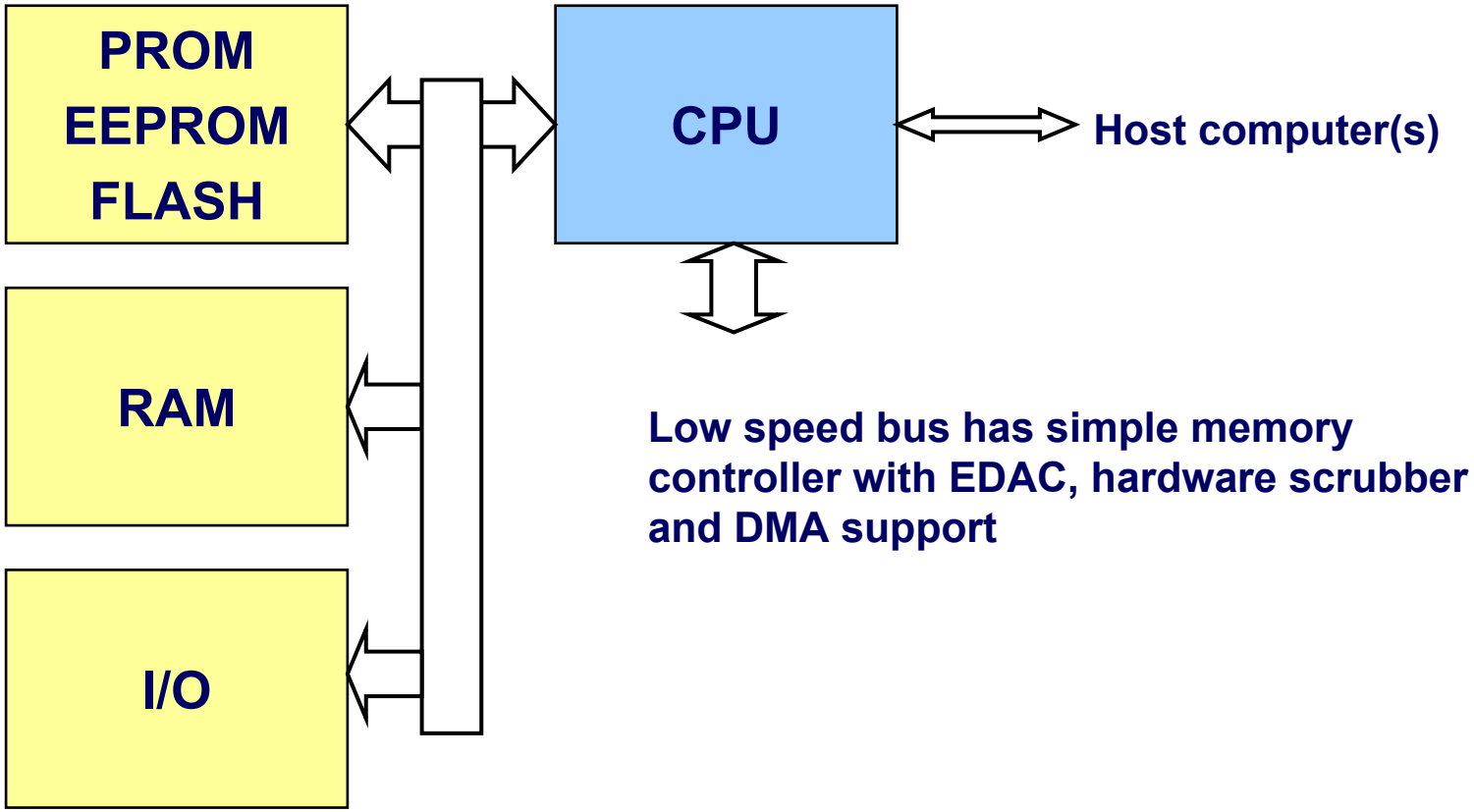
- **High performance system**



- Fast bus slave I/F available as IP
- No need for advanced memory I/F on the CPU

Possible hardware environments

- **Simpler system**



Low speed bus has simple memory controller with EDAC, hardware scrubber and DMA support


www.space.se