

Atmel SpaceWire activities status and plans

Dominique de Saint Roman Aerospace Marketing Manager Microcontroller Business Unit

Tel: +33 2 4018 1802

Mobile: +33 6 8037 0429

Dominique.de-saint-roman@nto.atmel.com



Agenda

- Rationales for going Spacewire
- Atmel technology roadmap
- Current IEEE1355 based products
- SMCS chips upgrade to SpaceWire standard
- SpaceWire plans
- Technology match
- Conclusions

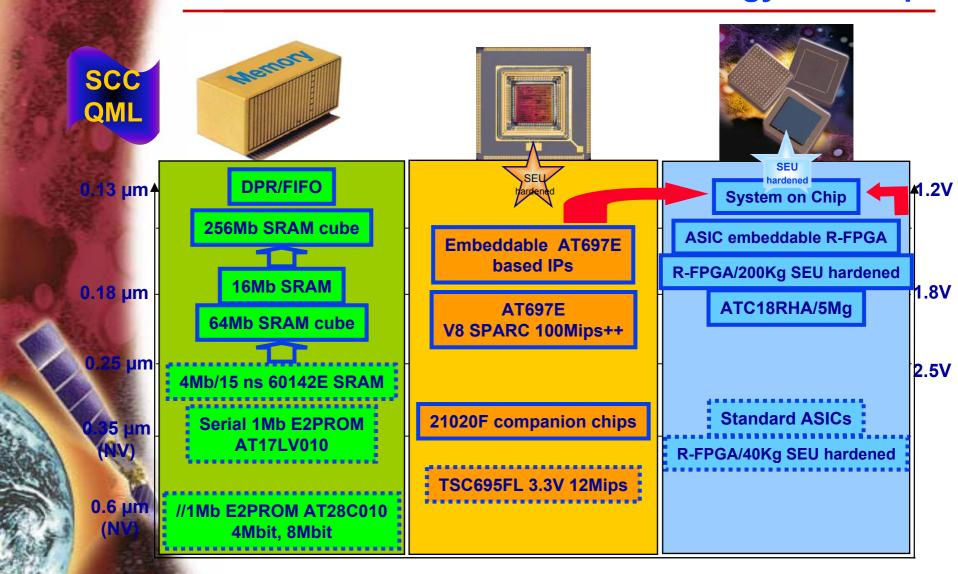


Rationales for going Spacewire

- First protocol communication activity to be actually coordinated:
 - > At the European level, with involvement of:
 - Space agencies
 - Space systems manufacturers
 - Universities and SMEs
 - Silicon manufacturers
 - Across the Atlantic ocean with NASA & JPL
- Standardization is the only means to achieve competitiveness, cost reduction, time to market and reliable designs
- Spacewire will help to reduce chip power consumption while the SoC trend is making chips more power hungry



Atmel technology roadmap





Current IEEE1355 based products

- Standard ASIC offering encompasses:
 - The TSS901E providing a triple point to point IEEE1355 based serial connections, so called « SMCS »
 - ➤ The T7906E providing a single point to point IEEE1355 based serial connection, so called « SMCS lite »
- Basic performances:
 - Latch up immune and 50Krads total dose capability
 - 200Mbps data rate (no LVDS)
 - Designed and industrialized prior to the SpaceWire joint efforts
 - Designed and application supported by Astrium Germany



SMCS chips upgrade to SpaceWire standard

- See Astrium presentation, but basically, the resulting chips will be:
 - pin compatible with former versions
 - produced on 0.5µm RT and RTP respectively for the triple and single point to point connection chips
- Initiative supported:
 - And co-funded by ESTEC
 - By Astrium through design and co-funding
 - By Atmel though industrialization and co-funding
- Tentative schedule:
 - Expected kick off in January 2004
 - Sampling in May 2004
 - Industrialization completed by September 2004
- Astrium / Atmel business model still to be agreed on



SpaceWire plans

- Atmel commitment is:
 - ➤ To industrialize the « SpW Router » standard ASIC UoD and Austrian are currently designing on our MH1RT ASIC library:
 - this raises the question of its adequation with the RTI supposedly to be designed on our 0.18µm ASIC library
 - ➤ To add the "SpW CODEC" to the relevant ASIC library (ies) once an agreement is reached with ESTEC as to how best achieve it (hard versus soft solution)
 - > Foreseen availability date for both: mid 2005



Technology match (1)

- Technically:
 - Both MH1RT & ATC18RHA have LVDS buffers
 - Power/speed performances:
 - MH1RT: 20mW / 215 MHz or 430 Mbps @ 3.3 V
 - ATC18RHA: 20mW / 325 MHz or 650 Mbps @ 3.3 V (simulated only, so far)
 - > MH1RT:
 - pre-diffused, resulting into lower NRE
 - 2 buffer sites per LVDS pair (TX and Rx)
 - > ATC18RHA:
 - pre-characterized, resulting into higher speed performances and NRE (!!!)
 - 2 and 3 buffer sites respectively per LVDS Rx and Tx pair



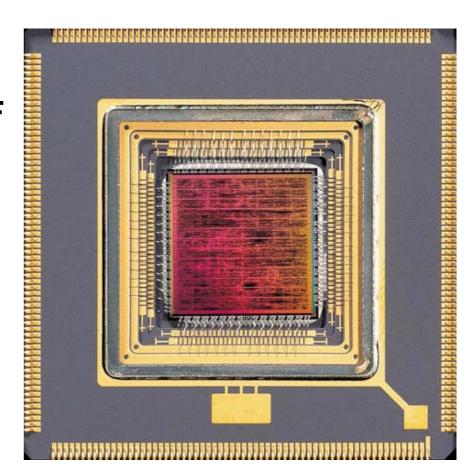
Technology match (2)

- Scheduling:
 - MH1RT fully released since 2001
 - > ATC18RHA:
 - Commercial ATC18 distributed since early this year
 - Alpha DK distribution on going
 - Beta DK released by end 1Q04
 - DK fully released by 4Q04



Technology match (3)

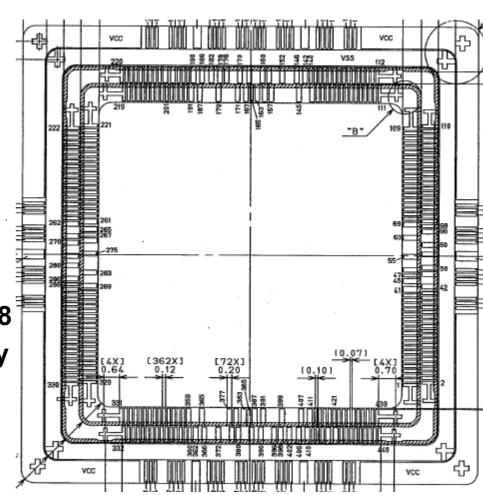
- Package is like a cable: therefore, high speed serial IF will need specific package layout
- The current cavities layout are already optimized for low resistivity path on logic bias pins, the so-called 4 decks cavity





Technology match (4)

- The new cavities will dedicate pins to LVDS in such a way that:
 - They are low L,R C
 - They have same // path
 - They are shielded
 - They are by cluster of 4/8
 - They are at the periphery for MCGA





Conclusions

- Atmel has the Si technology
- Atmel will have the products
- Atmel is actively preparing the environment for high speed IF
- Atmel has demonstrated its willingness to support the SpaceWire efforts
- Atmel is a partner for the space industry when going SpaceWire

THANK YOU!