# William Blake (1757-1827)

English painter,
engraver and poet …

… writes on asynchronous logic and systems-on-chip

**The hours of folly are measured by the clock;**
**but of wisdom, no clock can measure.**

**Proverb VI**

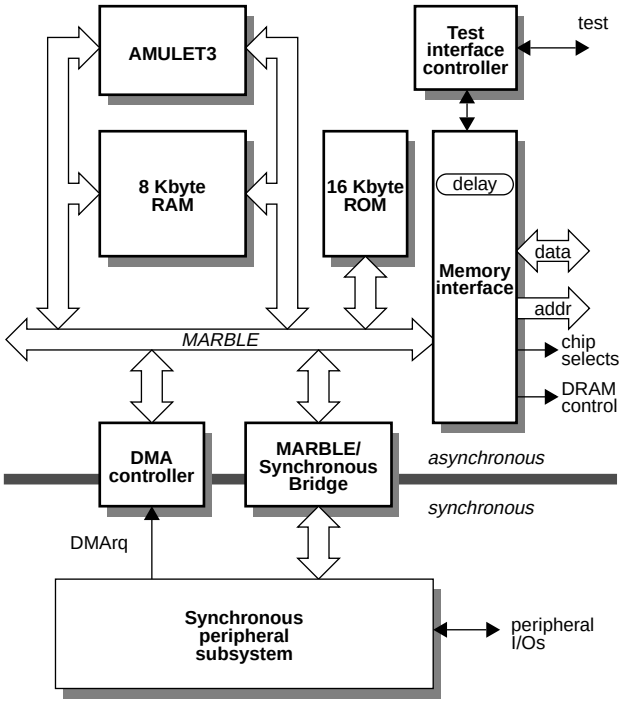
**I must Create a System, or be enslav'd by another Man's;**
**I will not Reason and Compare: my business is to Create.**

**The Words of Los**
**(From 'Jerusalem')**

THE UNIVERSITY of MANCHESTER

**AMULET group**

# AMULET3i - an Asynchronous System-on-Chip



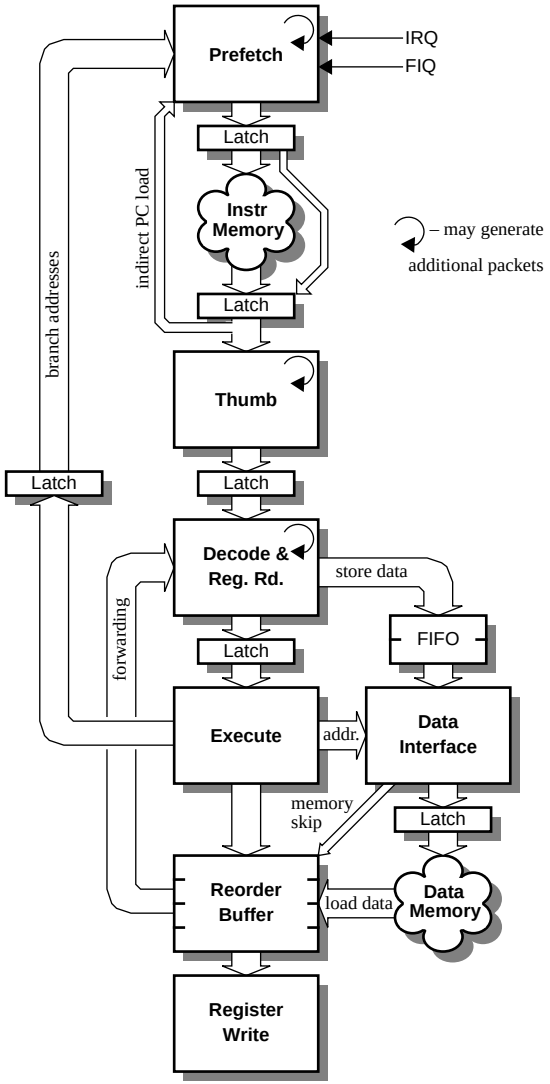

- AMULET3 microprocessor
- 8 Kbytes RAM
- 16 Kbytes ROM
- Flexible multi-channel DMA controller
- Programmable external memory interface
- MARBLE, a fully asynchronous on-chip bus
- Bridge to on-chip synchronous bus
- Configuration registers
- Software debug support
- Test interface

THE UNIVERSITY of MANCHESTER

AMULET group

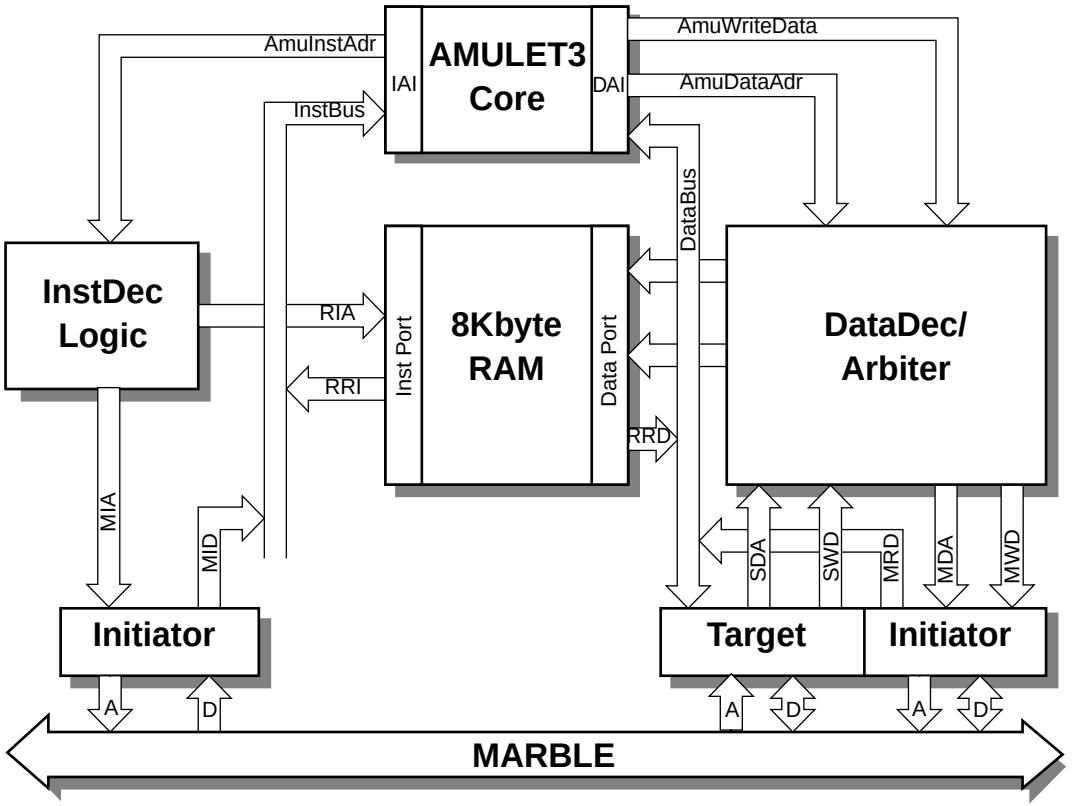
# AMULET3

- Familiar old AMULET core
- Relevant features here:
  - Two independent memory ports
  - Unified memory model
  - Reorders results, forwards loads
  - Can accept ‘late’ page faults



# Memory

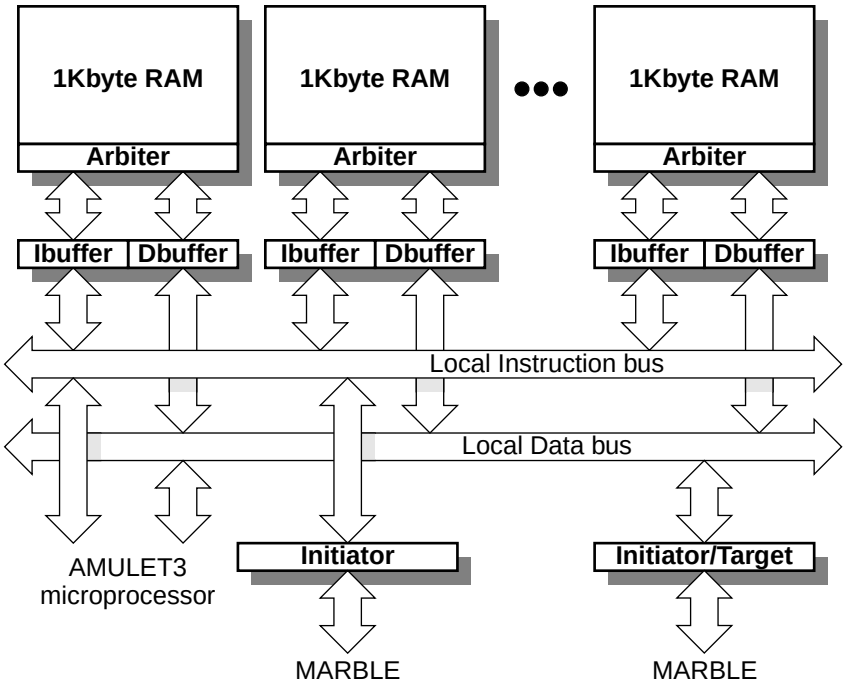
8Kbytes of RAM is accessible via two ‘local’ buses




- The RAM is ‘dual-port’ (at this level)
- The instruction bus is simpler so it has a higher bandwidth

# Memory structure

The local RAM is divided into 1Kbyte sub-blocks




- Unified RAM model
- Close to dual-port efficiency

Roughly half of instruction fetches are satisfied from the ‘Ibuffers’

# MARBLE

- Centrally arbitrated, multi-channel, asynchronous on-chip bus
- Separate, decoupled transfer phases for address and data
- Standard ‘master’ and ‘slave’ interfaces

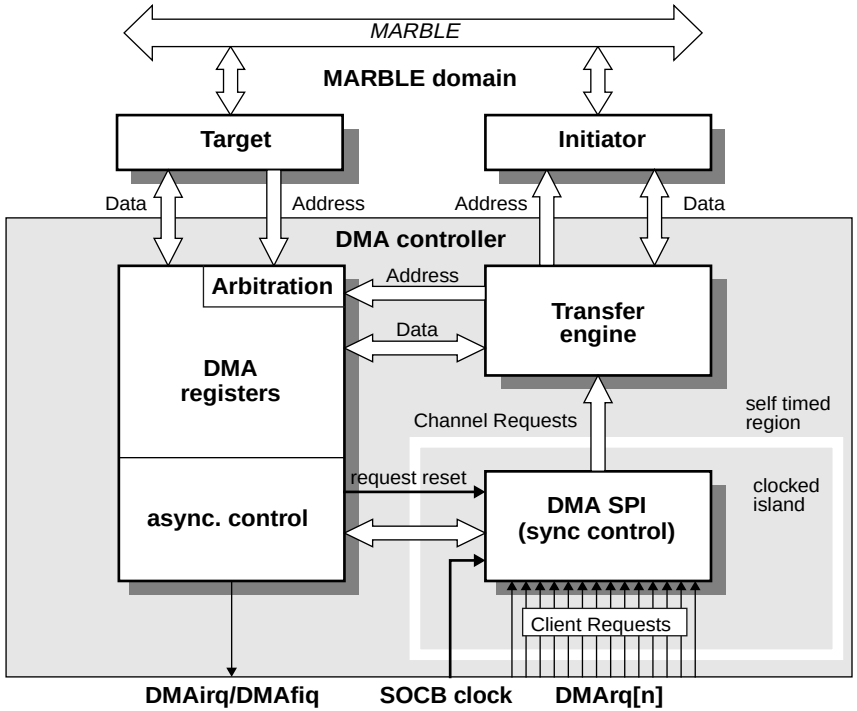
Supports

- 8-, 16- and 32-bit transfers
- bus locking
- sequential burst transfers

# Synchronous bridge

- A slave interface for clocked peripherals
- Performs synchronisation in the usual way (with usual risks)

# DMAC




- About 70 000 transistors
- Regular structures (i.e. register banks) in full custom design
- Control synthesised from Balsa desription
- Cheats slightly by letting a clock into one corner

# Balsa synthesis

- Balsa is a Tangram-like synthesis language
- The AMULET3i DMAC is the first substantial Balsa circuit

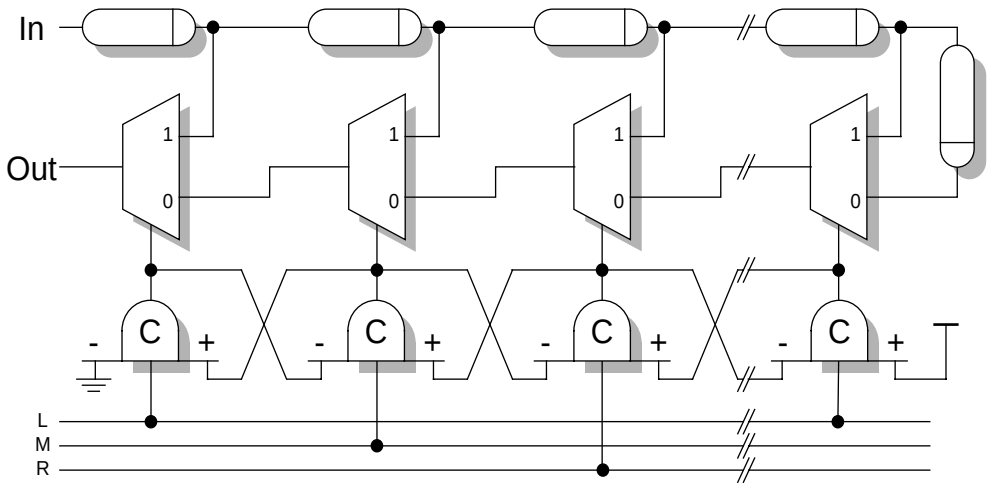
Balsa provides:

- Moderate performance
  - Easily enough to keep MARBLE saturated
- High designer productivity
  - The DMAC was basically Andrew's Balsa demonstator
- Easy modification
  - The specification kept changing, Balsa kept up

★★ Balsa is available NOW from your local AMULET stockist! ★★

# External interface

- Simple interface to external devices (as AMULET2e)
  - Includes DRAM support
  - Programmable in regions to be 8-, 16- or 32-bits wide (although pin restrictions disallow 32-bit option)
  - Timing support using novel on-chip delay chain (periodically calibrated in software)




- Test mode allows external access to system (via MARBLE)

# Vital statistics

## Transistor count

- AMULET3 – 113 000
- RAM (total) – 504 000
- DMA controller – 70 000
- EMI – 26 000
- Total – 800 000 (asynchronous subsystem)

## Geometry

- 0.35µm, 3 layer metal (using ARM's generic design rules)

## Area

- AMULET3i – ~25mm$^2$
- AMULET3 – ~3mm$^2$

Note: the local RAMs are relatively large in these generic, ASIC rules.

# Performance

- Peak Native MIPS 105 MIPS (110 in Thumb code)
- 176 kDhrystones[1]/s – 100 Dhrystone MIPS (ARM)
- 125 kDhrystones/s – 71 Dhrystone MIPS (Thumb) (-30%)

- AMULET3i power average 215 mW
  - 130 mW is within the processor core
- 465 MIPS/W for the system
  - 780 MIPS/W for the processor core

For comparison

- 0.35µm ARM9 ⇒120 MHz, (133 Dhrystone MIPS)
  800 MIPS/W

---

**1. Dhrystone 2.1 benchmark (normalised to VAX MIPS)**

# Performance Breakdown

These figures are based on a “typical” process and extracted layout using EPIC Timemill. Simulation conditions: 3.3 V, 25°C

## Processor core

- Instruction throughput is around 110-140 MIPS peak (no memory limitations, no load dependencies)
  - Register dependencies have no performance impact
- This compares favourably with ARM9 (120MHz on same process)
  - ‘Cycles/instruction’ generally the same too!

# Bus Speeds

## Local RAM bandwidths

Speed depends on bus and whether the ‘level 0 cache’ hits or not.

- Instruction bus ‘hit’ 9.5ns (105Mwords/s)
- Instruction bus ‘miss’ 12ns (83Mwords/s)
- Data bus ‘hit’ 13ns (77Mwords/s)
- Data bus ‘miss’ 16ns (63Mwords/s)

In typical code >50% of instruction fetches are ‘hits’.

## MARBLE

- Total bandwidth – 85Mword/s
- For any one initiator – 55Mwords/s

# Comments

- AMULET3i is about 2.5x faster than AMULET2e
  - The speed-up is about 1.7x when normalised for the different processes
- The performance is heavily limited by memory bandwidth
  - There should be another 30% here
  - (The designer changed continents too soon!)
- The Thumb decompression logic is a limiting factor in Thumb code
  - Speed was not a design priority here

# Testing

We still don't really know how to test it …

but:

- The external memory interface pins can be hijacked as a test interface (MARBLE bus master)
  - peripherals and memory can be read and written from a tester
- the processor can be used for Built In Self Test
- added features ease access to hard to test parts
  - e.g. the branch target buffer CAM is externally accessible

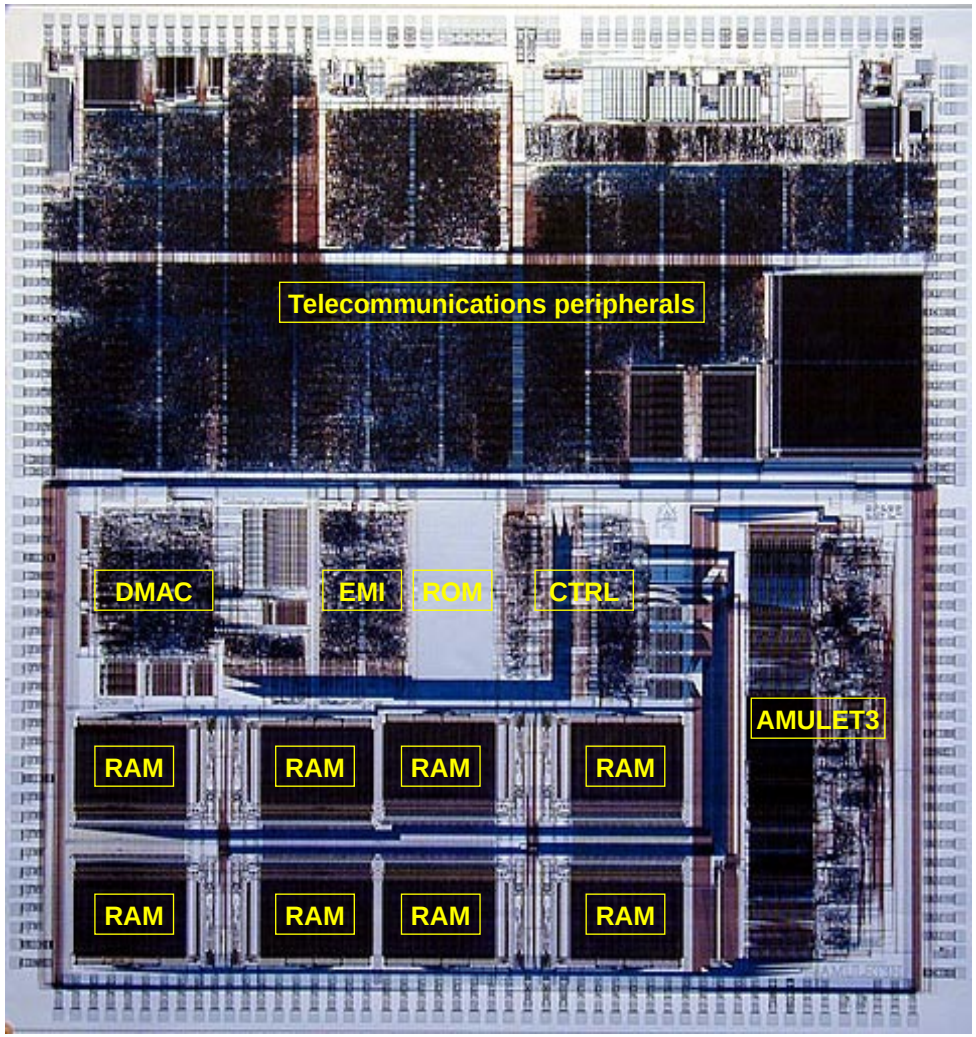
# Nice features (future expansion)

- Full virtual memory support (including ‘late’ aborts)
- Coprocessor interface support
- Debug break- and watchpoints in processor

# Nasty features (okay - “known bugs”)

- SWP not supported across MARBLE
  - MARBLE can be locked but the local bus does not expand SWP into two MARBLE cycles
- writes can complete following an earlier abort
  - another minor local bus problem

# DRACO




THE UNIVERSITY of MANCHESTER

AMULET group

# Conclusions

- Asynchronous logic is still a Good Thing

but

- Designing big systems is a lot of work

although

- With investment this technology can be competitive

- It's hard to keep up with the big companies

so

- Don't try this at home!

**www.cs.man.ac.uk/amulet/projects/AMULET3i.html**