chronous ALU", IFIP Working Conference on Asynchronous Design Methodologies, April 1993. Ed. Furber, S. B. and Edwards, M. D. Pub. North Holland.

asynchronous devices, namely the ability to adapt automatically to changing environmental conditions. The variation of performance and power-efficiency with voltage is shown in figure 2, using the Dhrystone benchmark as an indicator of performance and using the 1µm part. (The 0.7µm part operates at twice the speed but does not have the facility to measure core power consumption.) The voltage range used for these tests is limited by the other circuitry on the test card below 3.5V; the processor appears to operate in isolation down to 2.5V.

Variation of speed with temperature has also been measured. Here the test device displays a normal increase of delay of 0.3% per ˚C and operates correctly between -50˚C and 120˚C.

## 4: Future enhancements

AMULET1 is a first attempt at designing an asynchronous circuit of this complexity and, whilst the results from the first silicon are encouraging, there is considerable scope for improvement. The following enhancements are amongst those currently under consideration:

- Faster instruction decode logic. This is believed to be the critical section of the current design.
- Result and load data forwarding. Although conventional bypassing is not applicable in an asynchronous design, techniques have been found which offer the same benefits.
- Improved latches. Alternative latch technologies appear to offer improvements in both speed and power-efficiency.

The test devices present a micropipeline interface to external circuitry, requiring transition circuits to be constructed around external memory and peripheral components. This has presented some difficulties in the design of the test card and with hindsight was not a good decision. Although it is possible to build transition circuitry on GALs, the resulting circuits are very slow compared with on-chip cells. Indeed, with the 0.7µm part it is believed that the performance of the processor is limited by the speed of the test card. Versions of the AMULET processor currently under development will have more conventional interfaces to their external environment.

## 5: Conclusions

The design experience gained with AMULET1 demonstrates the feasibility of constructing a fully asynchronous microprocessor using conventional VLSI design tools. The resulting component is approaching the best synchronous design in power-efficiency and is not far behind in performance and area. As a system component AMULET1 is very flexible, automatically adjusting its performance to changes in temperature and voltage and using power only when there is useful work to be done.

There is considerable scope for improving both the power efficiency and performance of AMULET1. The area penalty for the asynchronous control logic is harder to overcome and will remain, though when comparing clocked and asynchronous organisations of similar complexity the overhead for the micropipeline design style has been estimated to be below 20% [5].

Future AMULET chips will employ an external interface which is much easier to design with, presenting the board-level designer with a relatively conventional task. The enhanced power-efficiency, particularly under highly variable load, will combine with the environmentally adaptive nature of the asynchronous circuitry to offer a highly flexible system component which will have significant technical advantages under suitable application conditions.

## 6: Acknowledgments

The AMULET1 design work described in this paper was carried out as part of ESPRIT project 5386, OMI-MAP (the Open Microprocessor systems Initiative - Microprocessor Architecture Project). The evaluation work has been supported as part of ESPRIT project 6909, OMI/DE-ARM (the Open Microprocessor systems Initiative - Deeply Embedded ARM Applications project). The authors are grateful for this support from the CEC.

The authors are also grateful for material support in various forms from Advanced RISC Machines Limited, Acorn Computers Limited, Compass Design Automation Limited, VLSI Technology Limited and GEC Plessey Semiconductors Limited. The encouragement and support of the OMI-MAP and OMI/DE-ARM consortia are also acknowledged.

## 7: References

[1] Furber, S.B., “VLSI RISC Architecture and Organization”, Marcel Dekker Inc., New York, 1989.

[2] Sutherland, I.E., “Micropipelines”, Communications of the ACM, Vol. 32, Number 6, June 1989, pp 720-738.

[3] Furber, S.B., Day, P., Garside, J.D., Paver, N.C. and Woods, J.V., “A Micropipelined ARM”, Proceedings of the IFIP TC 10/WG 10.5 International Conference on Very Large Scale Integration (VLSI’93), Grenoble, France, September 1993. Ed. Yanagawa, T. and Ivey, P. A. Pub. North Holland.

[4] Furber, S.B., Day, P., Garside, J.D., Paver, N.C. and Woods, J.V., “AMULET1: A Micropipelined ARM”, Proceedings of the IEEE Computer Conference, March 1994.

[5] Paver, N.C., “The Design and Implementation of an Asynchronous Microprocessor”, PhD Thesis, University of Manchester, June 1994.

[6] Paver, N.C., Day, P., Furber, S.B., Garside, J.D. and Woods, J.V., “Register Locking in an Asynchronous Microprocessor”, 1992 IEEE International Conference on Computer Design: VLSI in Computers & Processors. October 1992.

[7] Garside, J.D. “A CMOS VLSI Implementation of an Asyn-

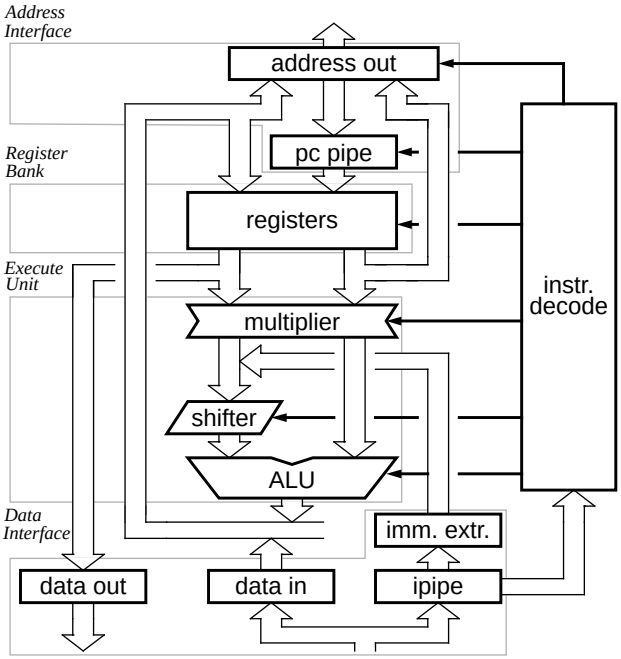


**Figure 2: AMULET1 organisation**

An unusual feature of the processor is the exploitation of an ALU with a data dependent evaluation time [7]. Circuitry within the ALU identifies the longest carry propagate path and the self-timing delay is adjusted accordingly. This allows a simple ALU structure to deliver very good average performance. Synchronous designs usually incorporate very complex adder circuitry in order to make rare worst-case operands complete within the clock cycle; asynchronous logic designers are freed from this constraint and can optimise the use of silicon resource towards typical cases.

The benefits of asynchronous design are also apparent in the multiplier circuit. In synchronous designs the clock is normally optimised for the carry-propagate adder in the ALU. Carry-save adders in a multiplier can operate much faster than this but a faster clock is unavailable, so instead several stages of carry-save adder are placed in series to fill the available time. This gives good performance at the cost of considerable silicon area. A self-timed multiplier effectively has its own clock which may be optimised for the local function, allowing high performance from a much reduced silicon area.

## 3: Device characteristics

AMULET1 was developed as a full custom design using Compass Design Automation tools, and has been fabricated on two CMOS processes: a 1µm process at ES2 and a 0.7µm process at GEC Plessey Semiconductors. Both devices have been evaluated on a test card which connects, via a serial line, to development tools from ARM Limited;

**Table 1: Characteristics of AMULET1 and ARM6**

| | AMULET1a | AMULET1b | ARM6 |
|---|---|---|---|
| Process | 1µm | 0.7µm | 1µm |
| Area ($mm^2$) | 5.5 x 4.1 | 3.9 x 2.9 | 4.1 x 2.7 |
| Transistors | 58,374 | 58,374 | 33,494 |
| Performance | 20.5 kDhry. | ~40 kDhry.[1] | 31 kDhry. |
| Multiplier | 5.3ns/bit | 3ns/bit | 25ns/bit |
| Conditions | 5V, 20˚C | 5V, 20˚C | 5V, 20MHz |
| Power | 152mW | N/A | 148mW |
| MIPS/W | 77 | N/A | 120 |

1. estimated maximum performance.

the monitor program in the test card ROM is the same as that used in similar evaluation cards for the ARM6. Both prototype devices are functional and execute programs produced by standard ARM development tools such as the assembler and C compiler. There are three minor design flaws which relate to the operation of interrupts and have relatively straightforward software work-arounds. A summary of the devices' characteristics is shown in table 1 with those of ARM6 for comparison.

The devices have been characterised over voltage and temperature variations and display the usual property of

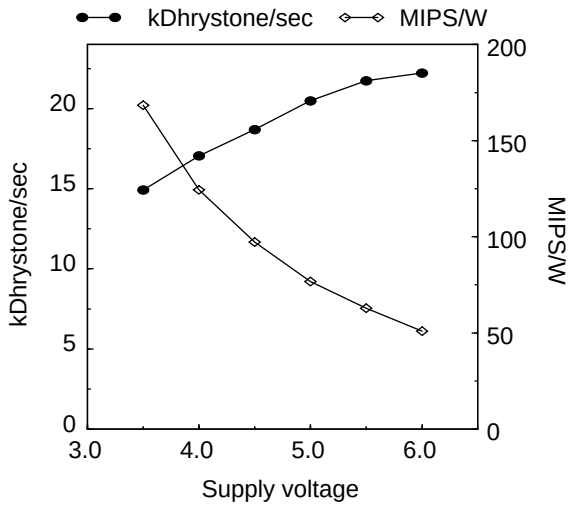


**Figure 3: 1µm AMULET1 performance and power-efficiency against voltage**

### 1.2: The ARM microprocessor

The ARM architecture [1] was chosen as the starting point for the AMULET work. The ARM is a leading processor for power-efficient applications and is relatively simple compared with other microprocessors of comparable performance. It has a dense instruction set for a RISC processor, which saves power by reducing memory bandwidth for instruction fetching and by increasing the effectiveness of a cache. Its low power is largely a result of the small transistor count and small cell area, though recent versions have been further enhanced through redesign to allow operation at low voltages.

## 2: Asynchronous design

Asynchronous design, whilst offering the potential advantages for power-saving outlined above, also introduces a number of new difficulties: where independent function units wish to communicate information, synchronization must be explicit; where independent units share a common resource in a non-deterministic fashion, arbiters capable of handling metastability reliably must be employed; the network of units comprising the complete system must be free from deadlock (and livelock).

The approach taken on AMULET1 is based on Sutherland's micropipelines [2]. Here all activity is *data driven*: a new data value arrives at a function unit in normal binary encoding on a bus; its presence is then signalled to the unit by a transition on the *Request* wire. This transition causes the unit to accept that data, signalling the acceptance by making a transition on the *Acknowledge* wire. The data can then change to its next value. The request-acknowledge signalling supports full flow control at every communications interface in the design.

Sutherland proposed a special form of 'capture-pass' latch for use in micropipelines which requires 24 transistors per bit. A revised form of this latch is illustrated in figure 2 where the transistor count has been reduced to 18 per bit. This is a 2-phase (transition sensitive) latch, where a transition on the 'capture' input (the C and nC complementary wires) latches the input data and a transition on the 'pass' input (P and nP) puts the latch back into transparent mode. These latches fit well into the 2-phase micropipeline framework, but have a high transistor cost when applied to 32-bit pipeline latches. On AMULET1 the decision was taken to employ conventional 4-phase (level sensitive) latches, identical to those used on ARM6, requiring 6 transistors per bit. This incurs an overhead in the control circuits for 2-phase to 4-phase conversion, but saves considerable area in the pipeline latches.

A decision which must be taken early in the design process is to determine the granularity at which concurrency is to be supported. A simple pipeline allows concurrent operation of each of its stages. The clocked ARM has a three stage pipeline, and therefore performs three concurrent activities in each clock cycle. An asynchronous micropipelined design can, in principle, support concurrent activity in each micropipeline stage. This suggests that the obvious way to implement a micropipelined ARM is as a three stage micropipeline with the memory system, the instruction decoder and the execution datapath being the three stages. However such a micropipeline structure would require every stage to operate for each instruction, offering little potential for improved power efficiency over the clocked design, so a finer grain of micropipeline was chosen.

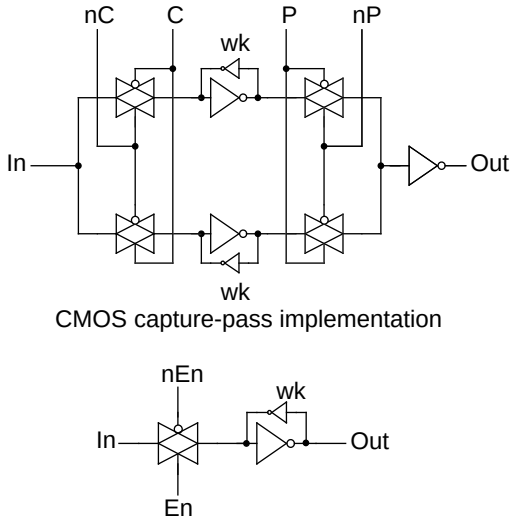


**Figure 1: 2-phase and 4-phase CMOS latches**

The processor is divided into four major functional units (figure 2) which operate concurrently and synchronize with each other only to exchange data. The data interface and execute pipe are relatively simple pipelines from the point of view of their control. The register bank incorporates a novel coherency mechanism [6] to handle read-after-write dependencies and the address interface autonomously issues instruction prefetch requests to a non-deterministic (but bounded) depth.

A major issue in any design which includes a pipelined execution path is the handling of read-after-write dependencies. Standard practice in synchronous design is to include register bypassing routes for dependencies of a small fixed number of clock cycle-lengths and register locks for dependencies of greater or indeterminate length. Bypassing depends on different pipeline stages operating in synchronism, and does not transfer readily to asynchronous pipelines. AMULET1 therefore does not incorporate bypassing, but rather relies on the register locking mechanism to control all read-after-write dependencies.

# The Design and Evaluation of an Asynchronous Microprocessor

S. B. Furber, P. Day, J. D. Garside, N. C. Paver, S. Temple and J. V. Woods

Department of Computer Science, The University,
Oxford Road, Manchester M13 9PL, England.

***Abstract***

*AMULET1 is a fully asynchronous implementation of the ARM microprocessor which was designed at Manchester University between 1991 and 1993. First silicon arrived in April 1994 and was found to be functional, demonstrating that asynchronous design of complex circuits is feasible with present day CAD tools.*

*This paper presents the motivation for the work, some of the design choices which were made, the problems which were encountered during the development of the design and the characteristics of the device itself. The future potential for asynchronous circuits is also discussed.*

## 1: Introduction

The growth in demand for high performance portable computing equipment has led to a resurgence of interest in asynchronous logic design techniques. In order to investigate the power saving potential of asynchronous approaches to CMOS design, a self-timed implementation of the ARM microprocessor [1] has been developed as a commercially realistic technology demonstrator.

The methodology applied to the design was based on Sutherland's "Micropipelines" [2], a bundled-data, bounded-delay model. Here, local timing signals are transmitted with a 'bundle' of data bits whose timing is constrained to ensure correct operation. This technique was chosen for its economy in silicon area and its potential for low electrical power consumption.

The organisation of AMULET1 has been described elsewhere [3,4,5] and details have been published on specific aspects of the design [6,7]. The focus of this paper is the design process and the behaviour of the test silicon. We conclude by speculating on the future potential for the commercial exploitation of asynchronous design techniques, basing these speculations on experience gained through the design and evaluation of AMULET1.

### 1.1: Power-efficiency

The introduction of RISC instruction sets in the early 1980s traded instruction semantic content for pipelined operation and RISC microprocessors have led the race for higher performance ever since. More recently, power-efficiency (measured in units such as MIPS per watt) has become as important as performance for significant application areas and processor designers have begun to turn their attention to power saving techniques.

One aspect of current microprocessor design practice which adversely affects power-efficiency is the use of a high-speed clock to control the operation of the pipeline. This clock forces the inclusion of large sections of logic which are normally redundant but which are required to ensure that rare worst cases complete within the clock cycle time. The clock also causes power to be dissipated in all the systems on a chip, whether or not they are performing a useful function at the time.

Modern power-sensitive design attempts to reduce this wastage by gating the clocks to independent function units so that they can be turned off when not needed. Such power management functions incur their own power costs, however, since they rely to some extent on software control, and running the power management software itself uses power; the more fine-grain the power management, the greater this software overhead becomes.

Asynchronous design effectively takes the granularity of the clock gating down to the lowest level by removing the clock altogether. It also removes the overhead of the management software and releases the designer from the constraint of worst-case design. For these reasons, asynchronous design has attracted attention as a technology with potential for low-power applications. The AMULET project at Manchester University was established to evaluate this potential in designs of practical complexity.