Differential register bank design for self timed differential bipolar technology

D.L.Jackson
R.Kelly
L.E.M.Brackenbury

Indexing terms: Differential RAM cell, Asynchronous design, Differential current-mode logic

Abstract: A high performance differential bipolar datapath based on the ARM architecture has been designed using 'micropipeline' self-timed techniques. The datapath design included a full-custom 31 × 32 bit register bank. Traditional bipolar single-ended design techniques are not suited to implementing a RAM of this size on the target technology. This has led to the adoption of a fully differential circuit for the RAM cell here. The paper describes the challenges of designing such a differential register bank and the surrounding self-timed control. The data path has been fabricated by GEC Plessey Semiconductors and is fully operational. Results for the register bank are presented in terms of speed, power consumption and area.

1 Introduction

The potential advantage of self-timed systems over their synchronous counterparts has created a resurgence of interest in 'asynchronous' design methodologies. Self-timed systems do not exhibit clock skew problems since there is no global clock. This is becoming an important consideration when large amounts of both silicon area and design time are dedicated to clock circuits and clock distribution, as shown in the DEC Alpha microprocessor [1].

Self-timed systems also have scope for increased performance. With fixed clock cycles, timing for synchronous designs are based on worst-case performance analysis. In self-timed circuits communication between blocks occurs when data is available. This enables self-timed systems to run at typical case performance rather than worst-case performance.

In addition, the innate modularity of self-timed systems makes for flexible designs and increases block reusability. This is becoming increasingly important, aiding high-level design and decreasing designers 'time to market' cycle.

Previous work had shown the feasibility of designing a complex microprocessor using self-timed techniques in CMOS [2]. The AMULET1 microprocessor was the first self-timed design based on a commercial 32 bit RISC architecture. AMULET1 runs code written for the ARM without the need for an external clock. The methodology applied was based on Sutherland's 'micropipelines' [3], 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, rather than a purely delay-insensitive model, was chosen for its economy in silicon area and its potential for low energy consumption.

As a follow up to this work it was decided to transform the CMOS AMULET design into high performance differential bipolar technology similar to ECL. This technology is marketed by GEC Plessey Semiconductors (GPS) as multilevel differential current mode logic (MDCML) [4]. MDCML is preferred to CMOS when performance is a primary consideration, since it is inherently faster than CMOS of a similar feature size and has a superior drive capability. However, it dissipates constant static power. Thus, although the MDCML design would no longer be low power, the overall performance was expected to improve by a factor of between 2–3 over that obtained by AMULET1.

2 Multilevel differential current mode logic

ECL was designed to overcome some of the limitations of other bipolar families by operating the switching transistors in the active region, avoiding stored charge, thus attaining higher switching speeds. To achieve this a differential pair is used as the basis of each gate, switching a current from one side of the differential pair to the other. In ECL the base of one of the switching transistors is supplied with a reference voltage, which determines the threshold switching voltage of the input signal at the base of the complement switching transistor.

MDCML consists of a constant-current, differential, digital circuit capable of operating up to 600MHz [5]. In this logic family all signal states are carried by *pairs* of wires and the actual state determined by the difference in voltage between these wires. The use of complementary signals allows a standard logic swing of only 160mV to be used and increases immunity to noise since it is now experienced as a common-mode signal

IEE Proceedings online no. 19971482

Paper first received 4th December 1996 and in revised form 4th June 1997

D.L. Jackson is with Cogency Technology UK, Bruntwood Hall, Cheadle, Cheshire SK8 1HX, UK

R. Kelly is with ICL (UK) Ltd., Wenlock Way, West Gorton, Manchester M12 5DR, UK

L.E.M. Brackenbury is with the Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK

which is rejected by the gate. Like ECL and CMOS, MDCML can be stacked into tree structures to give different levels of switching. GPS has chosen a three-level tree structure as the best compromise between power efficiency and complex gate functionality; this enables the gates to be operated from 3V.

Fig. 1 shows a simple three-input AND gate with the use of bilateral signal inputs and three distinct levels of switching.

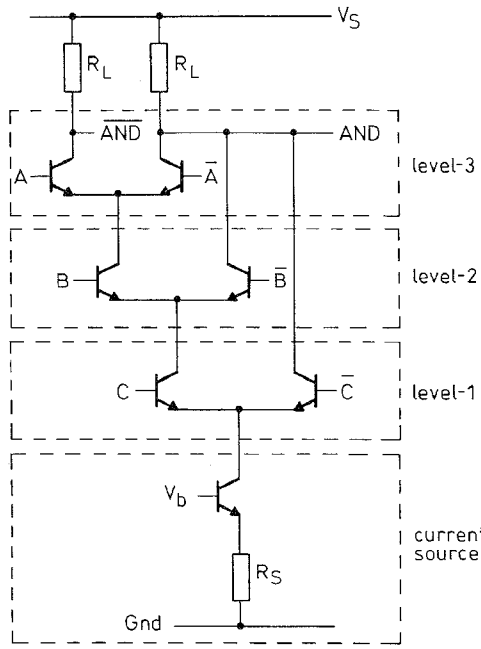

Fig. 1 *Three-Input MDCML AND gate*

3 System design considerations

The design of the ARM processor can be decomposed into a few major structural elements, one being the register bank. The ARM register bank contains 31 registers of which 16 are available to the programmer at a given time. All but one of these registers are general purpose and orthogonal.

The ARM architecture [6] defines a register based RISC processor in which arithmetic operations require one or two operands to be read from the register bank and a single result value to be returned. In existing synchronous implementations of the architecture, instruction execution is not pipelined (execution is a single stage of the Fetch – Decode – Execute pipeline) and an arithmetic operation is completed within a single clock cycle. In the asynchronous implementation instruction execution is decomposed into a number of pipeline stages. This concurrent execution improves performance but introduces the problem of data dependency.

Correct operation in a pipelined processor requires that data dependencies between instructions are respected; this may be achieved by ensuring that a location subject to modification cannot be accessed until the pending write operation has completed. This process is termed 'locking' and a novel arbiter-free method of maintaining dependencies was implemented in AMULET1 [7]. The method employed was a 'lock FIFO'. Here, the result register for an arithmetic instruction free to proceed is loaded in decoded form into a 32-bit wide first-in first-out buffer. When a result is returned from the ALU, it is paired with the buffer output to obtain the destination address. The item can then be deleted from the FIFO buffer. A similar FIFO buffer is used for external memory load references. Thus the FIFO buffers record all outstanding writes in the system.

Data dependencies are now detected by comparing source register addresses with valid entries in the FIFOs. A match indicates the source register is awaiting updating. In this case, the instruction is held until no match is detected from the FIFOs. All data in the FIFOs is held in decoded form as this leads to simple, replicated logic which yields a better performance than an encoded approach.

Since the register bank with lock FIFO and associated control was one of the largest blocks in the design, design effort to minimise the overall size and power was important. The register bank speed also makes a significant contribution to the overall performance. There is a trade-off between these factors and a compromise is required.

Most orders require at least one operand from the register bank. Due to area and power considerations, a single-port register bank had to be considered rather than the dual port bank used on all CMOS versions of the ARM architecture. In this case, two operand reads have to be processed sequentially. A representative mix of instruction types shows that typically dual reads are needed only 10% of the time and simulation leads to a predicted loss of performance of 7% for single port operation. However, most of this loss could be recouped by further reducing the requirement for dual reads by including a 'last result' register which stores the previous result of the ALU and which can be forwarded directly to the output of the register bank. A single-port register bank with sequential reads was therefore adopted as offering the best compromise between power/area and performance.

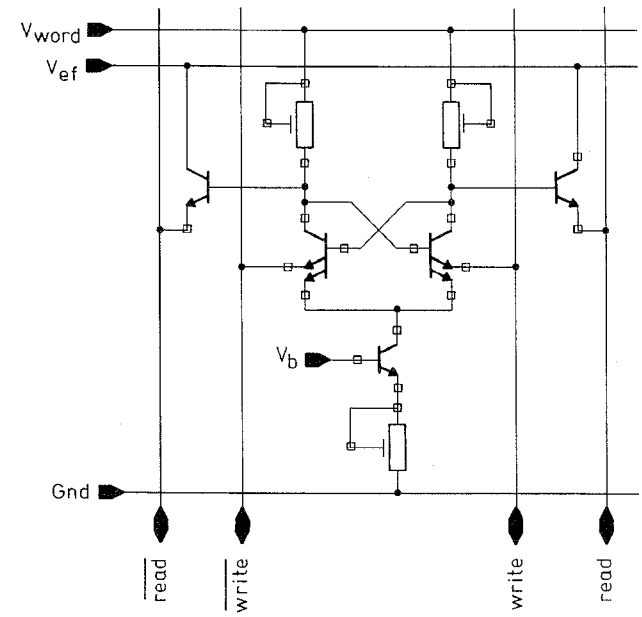

Fig. 2 *Bipolar SRAM cell*

4 Standard RAM cells

Traditional bipolar RAM cells are based on a cross coupled transistor pair and have differential bit lines but single-ended word lines. An example of a single port bit cell is shown in Fig. 2. To make a dual port version of this cell requires an almost complete duplication of the cell. Furthermore, this cell design is best suited to a synchronous framework where read and

write accesses do not occur concurrently. Since this timing cannot be guaranteed in an asynchronous system, arbiters would be required to separate read and write requests.

Previous work by GPS where standard RAM blocks had been mixed on the same die as differential logic had proved problematic. The bipolar technology employed in MDCML circuits uses a substrate ground plane. Ground connections are thus made by substrate taps with small equalising metal rails between taps. Variations in both the current density within the ground plane and the resistance of the substrate taps caused difficulties in single-ended word lines where the voltage levels are relatively small (~2V) and are determined with respect to ground. In MDCML, logic states are determined by the difference in voltage between two signal carriers and thus variations in the ground plane are less important. For all these reasons a differential framework for the register bank design was adopted.

5 Cell design

5.1 Memory cell

The differential memory cell used is shown in Fig. 3. The basis of a register cell is a standard differential bipolar transparent latch with a write enable (Wen) signal allowing data to be written into the cross-coupled transistor pair via a write bus (Wdat).

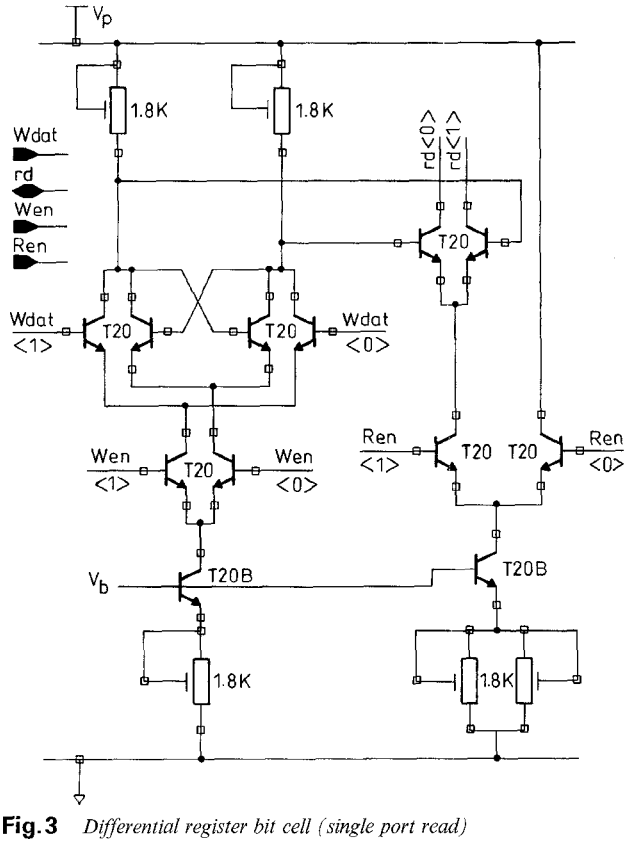

Fig. 3 *Differential register bit cell (single port read)*

The read bus can be seen as a type of tristate circuit. While the read enable (Ren) signal is not active, current is not drawn through either of the signal carriers and the bus floats in an undefined state. When a 'read' occurs, current is drawn through one of the signal carriers in each bit and forces a difference of voltage between the wires carrying the signal state of the bit. The load resistance for the read bus is not shown in Fig. 3 because cells do not have their own load resistance but the bus load is distributed through the cells.

By comparing Figs. 2 and 3 it can be seen that a consequence of opting for a differential design is an increased cell area because of extra components and additional external connections to the cell. A dual differential read port bit cell requires duplication of the read part of the cell including a rdB bus and a read B bus enable. This would lead to an increase over the size of a single port differential cell of around 25%.

It can be seen from Fig. 3 that writing and reading to cells is independent and can occur concurrently to different words. It is this observation which allows arbiter-free access to the cells of the register bank in an asynchronous environment.

5.2 Detecting read completion

In synchronous systems the register bank is assumed to have produced an output after a certain number of clock ticks. Asynchronous reads could be done in a similar way, using a matched bounded delay path for each read. However, the variable delay through the pipeline and the lock FIFO complicate the problem somewhat. To allow the register bank to begin a new read as soon as the previous one has completed, a read detect circuit is required.

Since the read bus floats when no read is occurring, a valid read is indicated when the value of the bus changes to a '1' or a '0'. Two read detect circuits are thus required on one bit of the bus. The circuit that detects a valid '1' is shown in Fig. 4. To detect a valid '0' the inputs to the circuit are swapped causing an effective input inversion. The outputs of the two read detect circuits are then ORed together to produce the control signal to indicate that a valid read has occurred.

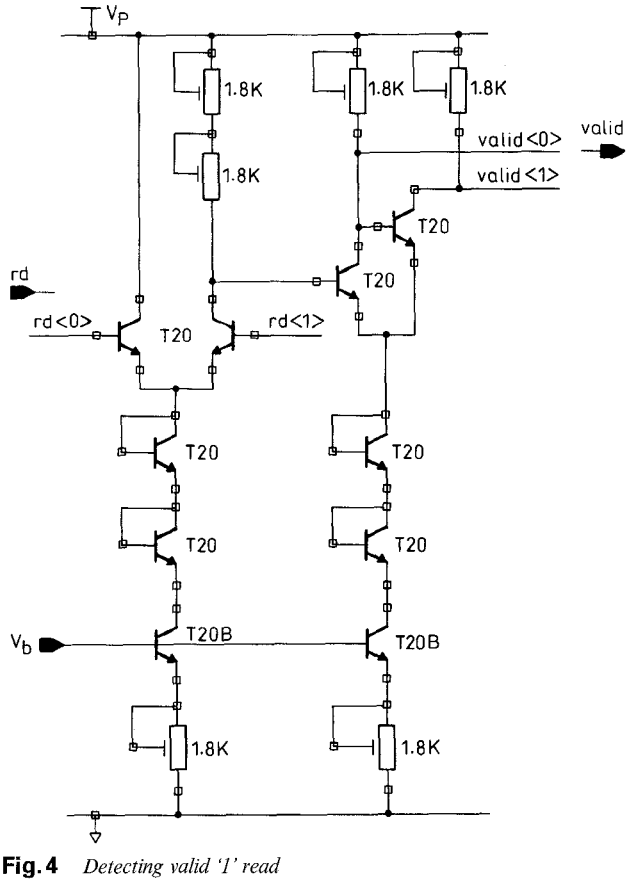

Fig. 4 *Detecting valid '1' read*

The circuit comprises a differential to single-ended double-voltage converter followed by a comparator. The physical closeness with which the components are placed means that the ground reference can be taken as uniform across the circuit so that single-ended operation here will not cause erroneous operation.

The comparator has an inbuilt threshold level halfway between that of a logic '0' and logic '1' at the top switching level. Voltages from the single ended converter are at one of three values corresponding to the bus floating or a read bus change to '0' or to '1'. In all cases, the voltage doubling means that the single-ended level is always distinctly above or below the comparator threshold. This allows conventional current switching between the transistors in the comparator switch tree to give a normal differential digital output. Only in the case of a read bus change to '1' does the comparator input exceed its internal threshold and indicates a valid '1'. For the other two cases, the output is '0' giving no valid indication. The valid detection is performed on the longest wiring path (bit 31) and so indicates the worst case timing.

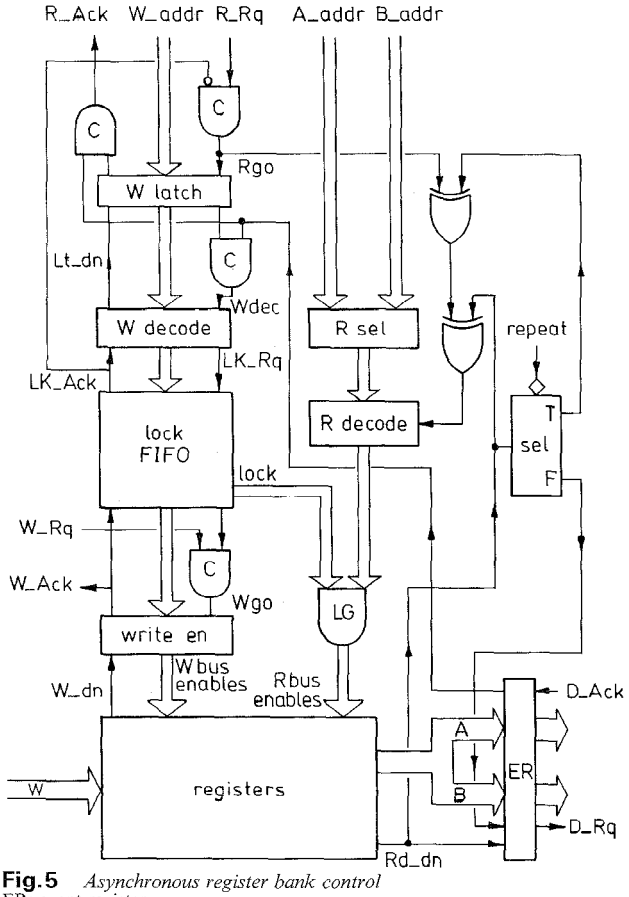

Fig. 5 *Asynchronous register bank control*
ER: event register
LG: lock gates

6 Self-timed control

An outline of the control framework for the register bank design is shown in Fig. 5. Externally and internally, a bundle of data is accompanied by two control signals; one (Rq) indicates to the receiver that the data is valid and the other (Ack), from the receiver, acknowledges the receipt of the data allowing the source to remove it. As two-phase signalling is used for the control signals, transitions on them indicate events and their levels have no significance. In such an event driven system, the timing and control signals are derived from the input control lines (R_Rq, W_Rq and D_Ack) and the 'read done' self-timing signal (Rd_dn). By the time these signals have propagated through the control elements accompanying the data paths, the data can be guaranteed to be valid, thus meeting the bundled data constraint which specifies that data must be valid before the accompanying control transition is sent.

Internally, a combination of two- and four-phase techniques need to be used for the control; the latter where a level is required e.g. the multiplexing of the A and B read address. These levels are generated from the appropriate two-phase signals.

The register bank is free to commence the next read operation (Rgo) when a read request (R_Rq) arrives and the preceding instruction has completed. If the read destination is locked, the request is stalled by the lock gates until it is unblocked. Every time an operand is read, the Rd_dn signal occurs causing the output event register to be loaded and the read decoder to be disabled. The select block is used to steer the Rd_dn transition to repeat the reading operation if a second operand is required. The final Rd_dn is steered to the false output of the select block iniating a request transition to the next stage (D_Rq). Locking of the destination address follows the acknowledgment of the read operands (D_Ack); locking must follow reading to prevent an instruction which reads and writes to the same register from stalling on itself. Once locking is complete, R_Ack indicates that the next instruction may be presented to the register bank.

Writing to the register bank is also asynchronous and independent of the reading activity. A write request (W_Rq) is paired with the output of the lock FIFO to determine the write address. Once written, the address is removed from the bottom of the lock FIFO and write completion indicated (W_Ack). The decoupling of read and write operations allows registers to be written and unlocked so that any instruction that is stalled will eventually be free to proceed.

7 Layout and size considerations

Although considerable area is saved by adopting a single-port register bank, area is still at a premium in MDCML designs. This mainly reflects the extra area required for differential connections at all points. For this reason and because a datapath largely comprises a bit replicated design (unlike its control), full custom design was undertaken for all datapath elements with control implemented in standard cell components provided by the manufacturer.

Both the register bank and lock FIFO are full custom designs. The register bank memory cell had a fixed height, determined by the pitch resulting from the ALU design. The cell height was determined not so much by the component area but by the internal connections and the number of differential highways (five) which needed to pass through each bit. These occupied around 40% of the 112.5μm pitch. As the cell height was fixed, the cell width was variable. Here, power and ground plus the input connections to the cell run orthogonal to the data flow, contributing to the cell width of 63μm.

The register bank layout is shown in the lower block of Fig. 6. Two 200μm power lines run the width of the bank, supplying power to each half of the bank. These can deliver a maximum current of 130mA each and

comfortably supply the bank current of 122mA. The compact layout of the register bank caused some problems with the lock FIFO layout since its height is essentially the register bank width; this is the upper block in Fig. 6. The area between the lock FIFO and register bank holds custom-designed power drivers. These were needed for the highly loaded signals into the bank. The same circuit design was used for both the enable controls from the lock FIFO and for the write bus, but here the layout had to be customised in each case to fit existing cell dimensions.

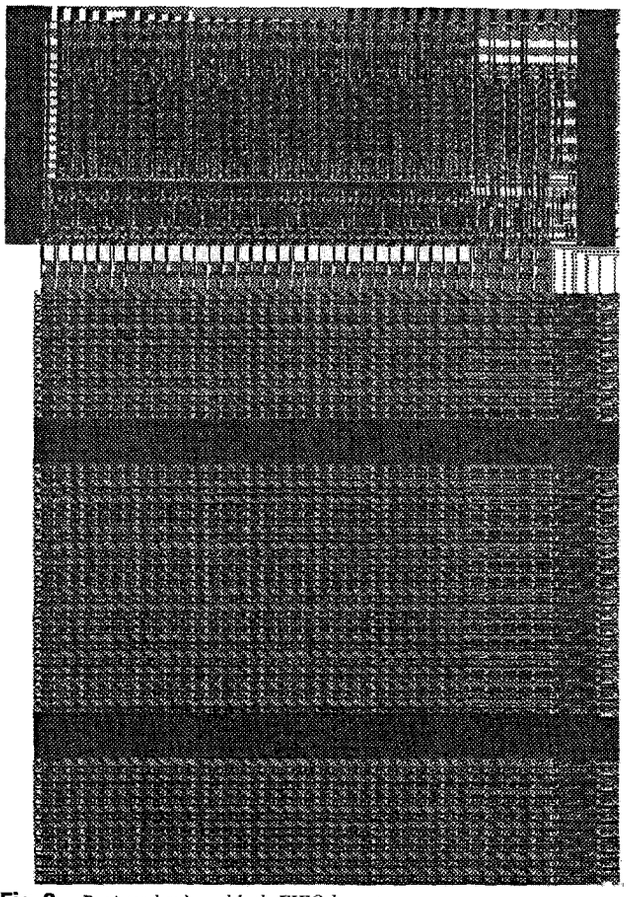

Fig. 6 *Register bank and lock FIFO layout*

Table 1 details the overall features of the datapath for the register bank and lock FIFO layout. This together with Fig. 6 convey the complexity and denseness of the design. The total number of components is equivalent to about 2250 matrix cells. In addition, the standard cell control for these blocks occupies 521 matrix cells. Given that the maximum array size in this technology is around 2550 cells in an 8 × 8mm chip, the necessity for a full custom datapath and a single read port for the register bank is clearly demonstrated.

Table 1: Register bank and lock FIFO datapath features

Block	Components	Size, mm^2	Power, mW
Register bank	21511	2.4 × 4.2	366
Lock FIFO	9204	1.6 × 2.7	156
Total	30715	2.7 × 6.2	522

8 Performance

A datapath comprising the register bank with the lock FIFO and an ALU has been fabricated. An overview of its architecture is shown in Fig. 7. The datapath is connected in a loop. Operands from the register bank pass via intervening pipeline buffer stages to the ALU. After the ALU operation, the result is written back to the register bank. The operations performed by the datapath are specified by the PLA section of the chip which contains a continuously cycling in-built test program.

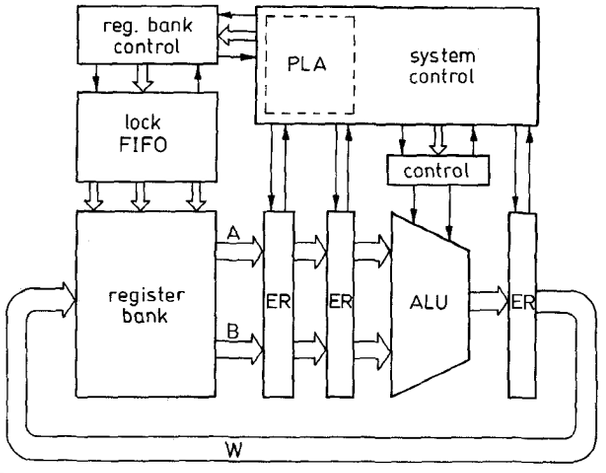

Fig. 7 *System architecture of fabricated chip*
ER: event register

The chip was placed in a test board which allowed its operations to be monitored; these could be internally or externally specified. The register bank performance has been obtained by measuring the time between successive incoming requests (R_Rq). If no operand read is requested on R_Rq then the time which elapses before the next request is received is 40ns. This represents the time for the register bank control to operate, recover and obtain the next request. Each operand read takes approximately an additional 30ns if the operand is not locked. Thus the time for a single nonlocked read was measured at 65ns and for a dual nonlocked read 100ns.

With the test program running on the architecture of Fig. 7, only a maximum of one locked register was ever observed. For a dual read, locking was only observed if the source register for the A bus was pending a write update. If the source register for the B bus was pending a write update then by the time the A-bus operand had been output from the bank, the source register for the B bus had been updated. The times observed for a locked operation were 96ns for a single read and 140ns for the dual read.

The performance of the register bank within the system shown in Fig. 7 is lower than expected. Instead of a factor of two increase in performance over AMULET1, the bipolar performance is slightly lower. Measurement of the basic gate delay shows that it is faster than the CMOS used on AMULET1 indicating that the MDCML chip could achieve a higher performance than has been obtained here.

The performance is attributable to the speed of the control rather than the datapath. Some speed loss in the control is due to additional gates inserted in the control to assist with the testability of the prototype design. Other losses are attributable to the overgenerous timing margins included in the control to compensate for the inability of our CAD package to back annotate designs containing a mixture of full custom and standard cell layout.

While the application of two-phase transition signalling has been successfully applied to MDCML technology, the resulting performance indicates that further work is necessary to establish more performance-efficient ways of implementing the control. This would include improving the two-phase transition timing through redesign, an investigation of four-phase self-timed control and examining whether synchronous techniques are more appropriate to this logic.

9 Conclusions

A full-custom register bank and lock FIFO in differential bipolar logic for use in a self-timed system have been described. Previous problems are known to have been experienced when integrating standard single-ended RAM parts onto a MDCML chip. This has caused a fully differential cell structure to be adopted for the design. The 1 Kbit register bank described operates correctly and reliably, justifying the decision to adopt a differential approach in a technology where logic swings are only 160mV and the supply rail is 3V.

Area considerations have proved to be an important feature of such a large design and has necessitated the use of a carefully handcrafted full-custom design and a single read port. The use of a novel read-detect circuit allows the operation to be self timed and operate within an asynchronous framework. The micropipeline approach has been successfully applied showing that the methodology is applicable to technologies other than CMOS. However, further work is required to enable control and data path times to be better matched.

10 Acknowledgments

This work formed part of the Transforming Architectural Models (TAM-ARM) project funded within the DTI/SERC Design Automation Programme and the authors are grateful for this support. The authors also acknowledge the help provided by their TAM-ARM partners - Advanced RISC Machines Limited and GEC Plessey Semiconductors. Current and past members of the AMULET research group in the Department of Computer Science at Manchester University are thanked for many stimulating discussions and suggestions.

11 References

1 DOBBERPUHL, D.W., WITEK, R.T., ALLMON, R., ANGLIN, R., BERTUCCI, D., BRITTON, S., CHAO, L., CONRAD, R.A., DEVER, D.E., GIESEKE, B., HASSOUN, S.M.N., HOEPPNER, G.W., KUCHLER, K., LADD, M., LEARY, B.M., MADDEN, L., McLELLAN, E.J., MEYER, D.R., MONTANARO, J., PRIORE, D.A., RAJAGOPALAN, V., SAMUDRALA, S., and SANTHANAM, S.: 'A 200-MHz 64-b dual-issue CMOS microprocessor', *IEEE J. Solid-State Circuits*, 1992, **27**, (11), pp. 1555–1565

2 FURBER, S.B., DAY, P., GARSIDE, J.D., PAVER, N.C., and WOODS, J.V.: 'AMULET1: A micropipelined ARM'. Proceedings of IEEE computer conference (Compcon'94), San Francisco, USA, March 1994, pp. 476–485

3 SUTHERLAND, I.E.: 'Micropipelines', *Comm. ACM*, 1989, **32**, (6), pp. 720–738

4 GEC Plessey Semiconductors: 'Differential logic design manual (FAB 4)'. 1.0 Edition, July 1988

5 GEC Plessey Semiconductors: 'ULA DX Series – high performance mixed signal array family'. January 1995, available as a PDF document at html://www.gpsemi.com/products/pdf/ds3746.pdf

6 FURBER, S.: 'ARM system architecture' (Addison–Wesley, 1996)

7 PAVER, N.C., DAY, P., FURBER, S.B., GARSIDE, J.D., and WOODS, J.V.: 'Register locking in an asynchronous microprocessor'. Proceedings of ICCD'92, October 1992, pp. 351–355