Reconfigurable Latch Controllers for Low Power Asynchronous Circuits

M. Lewis, J. Garside, L. Brackenbury
AMULET Group, Department of Computer Science
University of Manchester, Oxford Road
Manchester M13 9PL, UK

Abstract

A method for reducing the power consumption in asynchronous micropipeline-based circuits is presented. The method is based around a new design for latch controllers in which the operating mode of the pipeline latches (normally open/transparent or normally closed/opaque) can be selected according to the dynamic processing demand on the circuit. Operating in normally-closed mode prevents spurious transitions from propagating along a static pipeline, at the expense of reduced throughput. Tests of the new latch controller circuits on a pipelined multiplier datapath show that reductions in energy per operation of up to 32% can be obtained by changing to the normally-closed operating mode. Estimates suggest that in a typical application which exhibits a variable processing demand, a power reduction of between 16- 24% is possible, with little or no impact on maximum throughput.

1. Introduction

Reducing power consumption in integrated circuits is becoming increasingly important due to such driving forces as the demand for longer battery life and heat dissipation causing packaging difficulties. The absence of a clock in asynchronous circuits gives them some inherent advantages over synchronous circuits when designing for low power. A global clock causes power-consuming transitions throughout a circuit even when no useful work is being done. Also, clock distribution across a circuit can consume large amounts of power (for example ~1/3 in the DEC Alpha [1]).

One of the great advantages of asynchronous circuits is that it is possible to trade power consumption against processing speed dynamically. In applications where the processing demand is variable, it is possible to adapt the supply voltage in order to meet the throughput requirements at a given instant (Adaptive Supply Voltage Scaling or Just-In-Time Processing). An example has been demonstrated of a DCC error correction circuit [2] where an incorrect code word requires three times as much processing as a correct code word. As 95% of code words are correct, this allows a power saving of up to 80% by operating at reduced voltages during sequences of correct code words. Another application that has been demonstrated is a FIR filter bank for a hearing aid [3], where the supply voltage is reduced when processing low-level background noise. This kind of power saving strategy would be very difficult to implement in a synchronous circuit, as it would be necessary to reduce the clock speed to match the increase in circuit delay. The drawback to this method is the extra circuit complexity involved in varying the supply voltage.

The types of asynchronous circuits considered here are based around micropipelines [4], where timing of data transfers is managed at a fine grain by communication between adjacent processing stages. Fundamental to the design of micropipelines are latch controllers [5,6] which are responsible for negotiating data transfers between stages and passing data through at the appropriate time. One of the decisions that must be made when designing a micropipeline is how the pipeline latches are to be operated. If the latches are normally open (transparent) then data is passed down the pipeline as quickly as possible. However, any spurious transitions generated by the evaluation circuits are also propagated along the whole pipeline, wasting power. If the latches are normally closed (opaque), then these transitions are blocked at the expense of performance as the latches must be opened before data can propagate.

This paper describes a type of asynchronous latch controller whose operating mode can be selected by means of a control signal. This allows power consumption to be traded against speed with only a small increase in circuitry compared to the fixed normally-open designs. During periods of low demand, power consumption is minimised by selecting the normally-closed operating mode. This represents a significant power reduction: a study of the synchronous ARM 6 processor [7] has shown that a power saving of 35% could be achieved in the ALU simply by using a normally-closed latch on its input.

2. Protocols in Asynchronous Circuits

The latch controllers considered in this paper use a four-phase communication protocol, based on Request and Acknowledge signals, as shown in Figure 1.

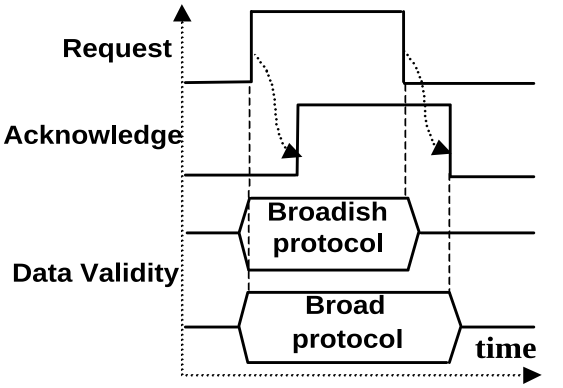

Figure 1. Four-phase Handshake Protocols

The request signal indicates that data is ready from the preceding stage. Once the data has been latched, the acknowledge signal is sent back to allow the previous stage to release the data and move to the next item to be processed. Part of the protocol definition specifies when the data is to be stable (that is, when the sending latch is guaranteed to be closed). Latch controllers implementing two types of protocol are considered in this paper. In the "broad" protocol, the data is held stable from the assertion of Request to the removal of Acknowledge. In the "broadish" protocol, the data is held stable only during the period that Request is active. These protocols are in common use in asynchronous systems as they allow the Request signal to also be used as an enable signal for processing logic.

The latch controller opens and closes the data latches at the appropriate points in time, depending on the protocol being used and the operating mode. These operating modes are shown in Figure 3, with the notation that when enable is high, the latch is open. The extra transition required to capture the data in normally-closed operation slows down the response of the handshake circuit. This means that the normally-open operation is chosen where operating speed is the most important criterion. However, normally-closed operation guarantees that no spurious transitions are propagated through the pipeline, as the data is defined to be stable during the period when the latch is allowed to open.

These circuits can then be built up into pipelines with processing logic between the sending and receiving latches. Timing is managed through either a completion signal from the processing logic or a matched delay in the handshake path. A typical pipeline structure is shown in Figure 2.

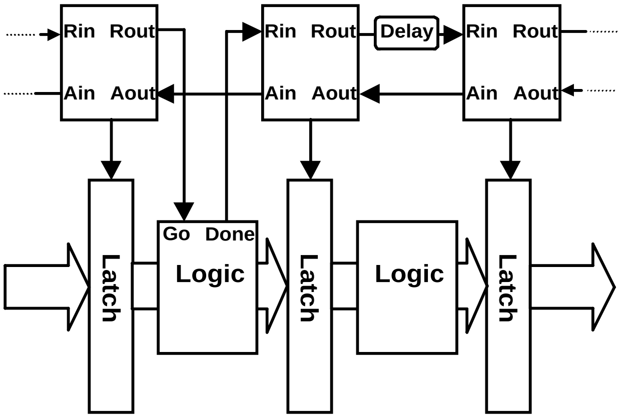

Figure 2. Asynchronous Pipeline with Processing

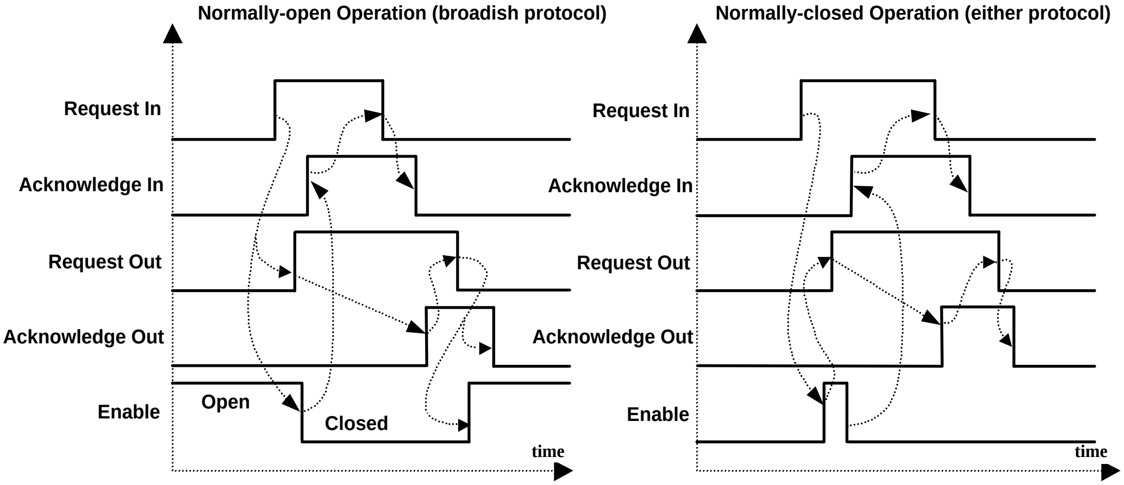

Figure 3. Latch Controller Operating Modes

3. Reconfigurable Latch Controllers

In many applications, a high maximum throughput is required but this maximum throughput is only needed for small periods of time, with periods of lower load between them. This introduces the possibility of dynamically reducing power consumption during periods of low load at the expense of reducing throughput, such as in the adaptive voltage scaling schemes described previously. A new form of latch controller circuit has been developed, based on the original normally-open latch controller designs by members of the AMULET group [9]. These allow the operating mode of the pipeline latches to be selected by means of an external “Turbo” signal. When maximum throughput is required, the Turbo signal is made high and the latch controller circuit operates in normally-open mode. When the circuit is less heavily loaded, the Turbo signal can be brought low. The latch controller circuit then operates in normally-closed mode and spurious transitions are blocked. The modified latch controller circuits are shown in Figure 4 and Figure 5. The gates labelled 'C' are Muller C elements, with asymmetrical inputs indicated by a '+' or '-' sign (affected only by high or low inputs respectively).

Operation of Broad Protocol Latch Controller
When idle, Na is high and the state of the latch enable line is controlled by the Turbo line. The operation of the latch controller on receiving Rin then depends on the state of Turbo.

If Turbo is low (normally-closed operation), receiving Rin causes the latch to open. Once the latch is open, the left-hand C gate drives Na low which causes the latch to close and capture the data, and the right hand C-gate to drive Rout high. Once the latch has finished closing, Ain is asserted. Rin going low then resets the input half of the circuit, and ensures that the latch remains closed. When Aout is received, the right hand C-gate removes Rout.

If Turbo is high (normally-open operation), receiving Rin immediately causes Na to be driven low, causing the latch to close and the right hand C-gate to drive Rout high. When the latch has closed, Ain is asserted, and Rin going low resets the input as above. Either Rout or Aout being high holds the latch closed, and Rout is reset by receiving Aout. The left hand C-gate ensures that the latch be released before another Rin can be acknowledged.

Operation of Broadish Latch Controller
The operation of this circuit is very similar to that of the broad protocol controller. The main difference is that, in this case, it is only Rout being high that holds the latch closed. When Aout is received, the right hand C-gate removes Rout which then gives control of the latch to the Turbo input and allows the input stage to reset with Rin going low.

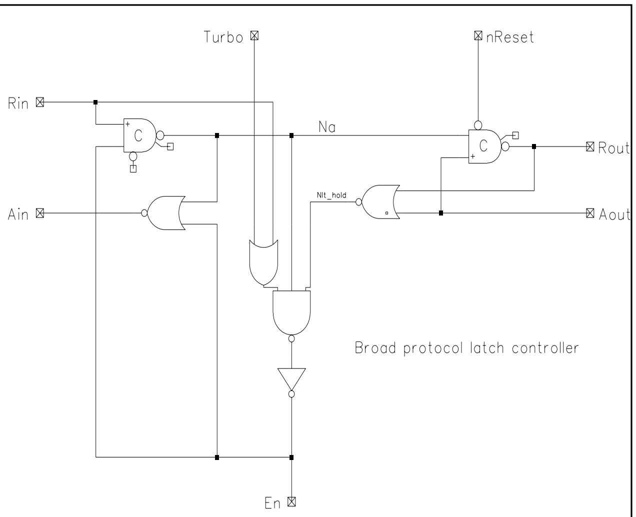

Figure 4. Broad Protocol Reconfigurable Latch Controller

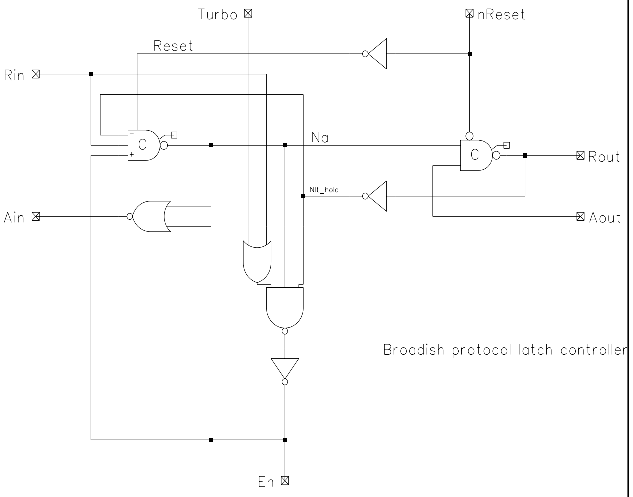

Figure 5. Broadish Protocol Reconfigurable Latch Controller

4. Verification of New Designs

The key requirement of asynchronous control circuits is that they should be speed-independent, that is they should operate correctly regardless of the delays of individual circuit elements. The reconfigurable latch controllers are based on conventional normally-open latch controllers that have been used previously. The first step in the evaluation of the new latch controller circuits was to verify that the circuits are still speed independent and that they implement the four-phase protocols correctly using the Petrify tool [8]. To verify the circuits, signal transition graphs for correct operation were derived. Petrify was then used to synthesise speed independent circuits and the results compared with the original circuits.

The latch controller circuits are required to synchronise three distinct sequences: the input Request / Acknowledge cycle, the latch open and close cycle, and the output Request / Acknowledge cycle. The time sequences for these cycles are shown in Figure 6.

The cycles have to synchronise at different points, depending on whether the broad or broadish protocol is being implemented and in which mode the latches are to be operated.

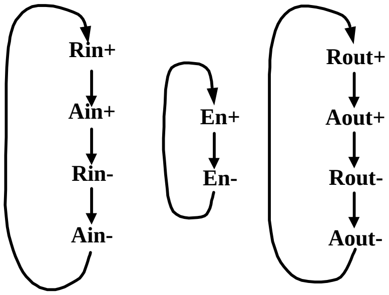

Figure 6. Latch Controller Cycles

In order to allow for concurrency between the input and output cycles it is necessary to add an internal state variable, Na (as in Figure 4 and Figure 5). Na goes low to indicate that a Request in has been accepted and Acknowledge is pending. STGs for normally-open operation of the broad and broadish latch controllers exist already [9] and are shown in Figure 7. It can be seen from the signal transition graphs that the reopening of the latches (En+) occurs immediately at the appropriate point of the protocol.

For normally-closed operation, it is desired that the latches remain closed until Rin is received. This can be done in a straightforward manner by moving the arc between Rin+ and Na- so that it goes instead between Rin+ and En+, as shown in Figure 8. The operation will then proceed correctly, with the latch opening and then closing on receipt of Rin+.

The primary difference between the STGs for the two operating modes is the position in the graph where the latch is enabled. It is a relatively straightforward task to combine the STGs for normally-open and normally-closed operation by drawing a Petri net with a choice of paths depending on the state of the external Boolean control signal Turbo. In order that circuits could be synthesised from the final Petri net, it was necessary to add conditions on the Turbo signal changing state as well. This requirement corresponds to the hazard that would be caused by Turbo changing state during the period between Rin-Ain cycles, when the state of the pipeline latches is dependent solely on the state of the Turbo signal. As the data in the pipeline is not guaranteed to be stable at this point, closing the pipeline latches risks metastability. The precise setup and hold times on the data for changes in the Turbo signal are dependent on the setup and hold times of the pipeline latches and the propagation delay from the Turbo input to the latch enables.

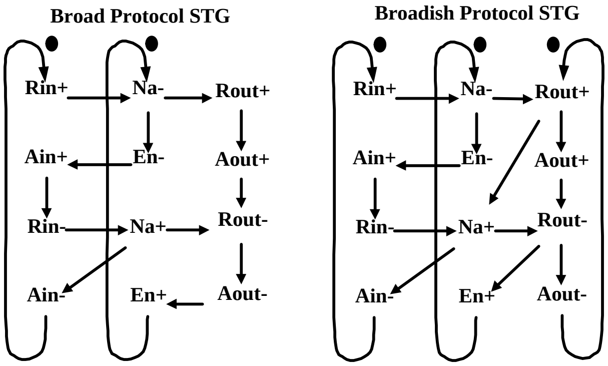

Figure 7. Broad and Broadish Protocol STGs for Normally-Open Operation

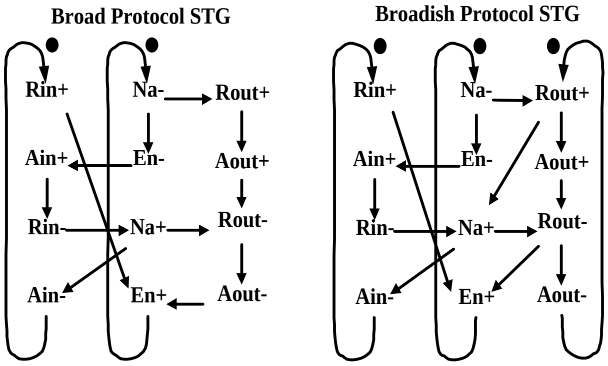

Figure 8. Broad and Broadish Protocol STGs for Normally-Closed Operation

From examination of the circuits in Figure 4 and Figure 5, it can be seen that the Turbo signal passes through an OR gate with the Rin signal. Whenever Rin is high, the output of this OR gate will be independent of the state of Turbo. Changes in the Turbo signal will be isolated from the rest of the circuit, and the latch enables in particular. It was therefore chosen that Rin being high should be the condition for changing Turbo. On putting this condition into the mixed STG / Petri nets, it was found that speed independent circuits could be synthesised and that they were identical to the original designs.

5. Controlling the Turbo signal

As discussed previously, it was necessary to form timing requirements on when the state of the Turbo signal can change. These timing requirements mean that it is not possible to have a global Turbo signal. Instead, Turbo is passed down the pipeline by edge-triggered latches clocked by Rin as shown in Figure 9.

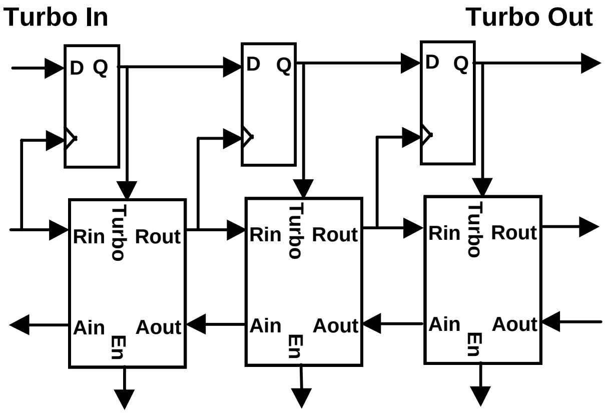

Figure 9. Modified Pipeline with Turbo Signal

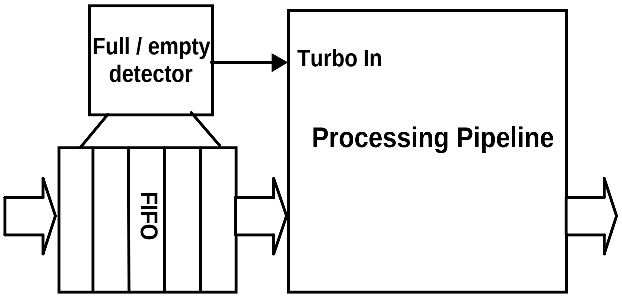

Figure 10. Controlling the Turbo Input According to Demand

It should be noted that this is not a truly speed-independent way of passing the data, as it is theoretically possible for the previous latch controller in the pipeline to move onto the next input cycle before Turbo can be captured, thereby risking metastability. It is therefore necessary to ensure that the time taken to capture Turbo from Rin going high is much less than the time taken for the Rin - Ain cycle of the previous stage. In practice, this is likely to always be the case as the Rin - Ain cycle is slowed by the processing delay of the previous stage.

It is necessary to develop a scheme to decide the state of the Turbo signal at any given time. One possible example of how the choice of Turbo signal could be made is shown in Figure 10. In this example, the occupancy of a FIFO buffer is used to determine the operating mode of the latches. This would be suitable for a DSP type application where an operation is performed on an incoming input stream. An alternative way of controlling Turbo, in a microprocessor for example, would be to have it under software control. When the system was switching into a high demand state (for example, going from idle to call mode in a digital cellular phone system) a Turbo bit could be set and allowed to propagate throughout the system.

6. Test Circuit Design

In order to test the new latch controller circuits, the designs were incorporated into a substantial circuit consisting of a pipelined 32x32 bit multiplier datapath. The block diagram of the multiplier is shown in Figure 11. The circuit has a five-stage pipeline and should therefore show a clear difference between normally-open and normally-closed operating mode. Multiplication is a typical operation in digital signal processing, an area where low power is becoming increasingly important for applications such as cellular phone handsets.

The circuit is based around arithmetic elements of the AMULET3i processor [10]. Full-custom layout for the datapath was chosen, as the layout cells were already available, in order to provide more accurate results. Interconnection delays are becoming increasingly significant as design rules are scaled down. A full layout simulation displays timing behaviour significantly different from a circuit simulation that does not take these interconnection lengths into account. The generation and propagation of glitches is critically dependent on timing, so that precise simulations are essential for accurate power estimation.

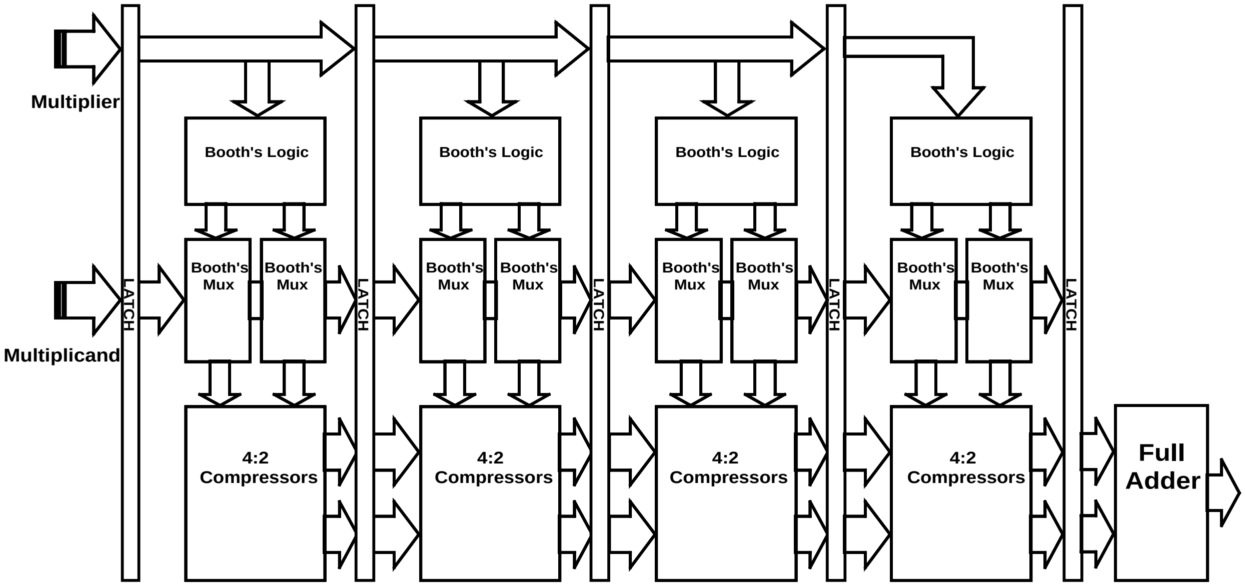

Figure 11. Multiplier Datapath Structure

Booth encoding of the multiplier is used to halve the number of additions required, and redundant carry-save addition is used at each stage. The partial sum and carry is resolved using a full adder at the end of the pipeline. The design uses the Booth multiplexer, and 4:2 compressor layout cells and also the full adder from the AMULET3i ALU. Eight bits of the multiplier are encoded at each pipeline stage. Timing at each stage is managed by an extra stripe in the datapath which mimics worst-case data along the critical path of the stage.

7. Tests Performed

The main parameters under test were the differences in energy efficiency and throughput between the configurable latch controllers in normally-open and normally-closed operating mode, and the conventional normally-open latch controllers on which they are based.

The power wasted by the passage of spurious transitions is strongly dependent on the pipeline occupancy. If the pipeline is partially full then any spurious transitions will be blocked at the first full stage. To investigate this effect, a C language model was written to send input data so as to try and maintain a certain level of pipeline occupancy. The model was also designed so that the arrival of the multiplier and multiplicand could be skewed in time.

EPIC Timemill was used to analyse the throughput of the pipeline using each latch controller. The throughput was measured with different pipeline occupancies, in each case operating the reconfigurable latch controllers in both normally-open and normally-closed mode. Random data was used for the multiplier input as the performance is not data-dependent.

EPIC Powermill was used to provide the relative comparisons of energy consumption in this study.

Tests were performed with each type of latch controller for different levels of pipeline occupancy. The effects of skewing the inputs were investigated for the case of a pipeline occupancy of one. Again, the reconfigurable latch controllers were tested in both normally-open and normally-closed mode and were compared in their performance against the conventional normally-open latch controllers.

Power consumption is strongly data dependent, and so two separate sets of tests were performed. The first set was run with random input data, while the second set used simulated data from an 8-pole FIR low-pass filter operation on an excerpt of sampled speech. Data sets from real applications display correlations between successive values which can strongly affect power consumption due to reduced numbers of changing bits, and can also have sections of low level signals which exhibit frequent sign changes, possibly increasing power consumption. A filtering operation is typical of the input to the multiplier in DSP applications.

8. Results

8.1. Throughput Tests

The throughput results in Figure 12 show an increase in throughput up to an occupancy of three, after which no further increases occur and the multiplier pipeline operates at its maximum speed.

The broad protocol configurable latch controller shows an increase in throughput of 6.8% for the normally-open case over the normally-closed case when the pipeline is operating at maximum throughput. The difference is greater when the pipeline was operating with lower occupancy, with an increase of 12.4% when operating with a single item in the pipeline at any time.

The broadish protocol configurable latch controller shows an increase in throughput of 7.8% for normally-open mode over normally-closed when operating at maximum throughput. The increase in throughput when running with single items in the pipeline is 12.5%.

It is to be expected that the configurable latch controllers would have a slight performance penalty over the conventional latch controllers. The configurable latch controllers have an extra input to the NAND gate controlling the latch enable signal. This requires a pair of extra transistors in the gate tree, and also implies extra capacitance. It was found that the broad protocol configurable latch controller had a penalty in maximum throughput of 4.2% when compared to the fixed normally-open latch controller. However, the broadish protocol configurable latch controller had a negligible difference in maximum throughput. The broadish protocol allows for the latches to be freed up before the Acknowledge cycle has completed at the output. This overlap improves the performance of the broadish protocol when operating at maximum capacity, and hides the reduced performance due to the configurable latch controller circuit

8.2. Power Consumption Tests: Random Data

The graphs of energy consumed per operation against pipeline occupancy in Figure 13 show that the power consumed using normally-open latch controllers decreases as the pipeline progressively becomes more full. This is to be expected from the simple model of spurious transition propagation; transitions can propagate less far as the pipeline becomes more congested. It is interesting to note that using normally-closed latch controllers causes the energy consumed per operation to become virtually constant.

When using the broad protocol configurable latch controller the energy per operation decreases by 1.8% in normally-closed mode, as compared to normally-open mode, when running at maximum throughput. When operating with a single input value at a time, the difference between the operating modes becomes much more significant, with an increase in energy per operation of 21% for normally-open mode over normally-closed mode. The difference for a single input value becomes even greater when the multiplier and multiplicand inputs are also skewed in time, giving a 32% increase in energy per operation.

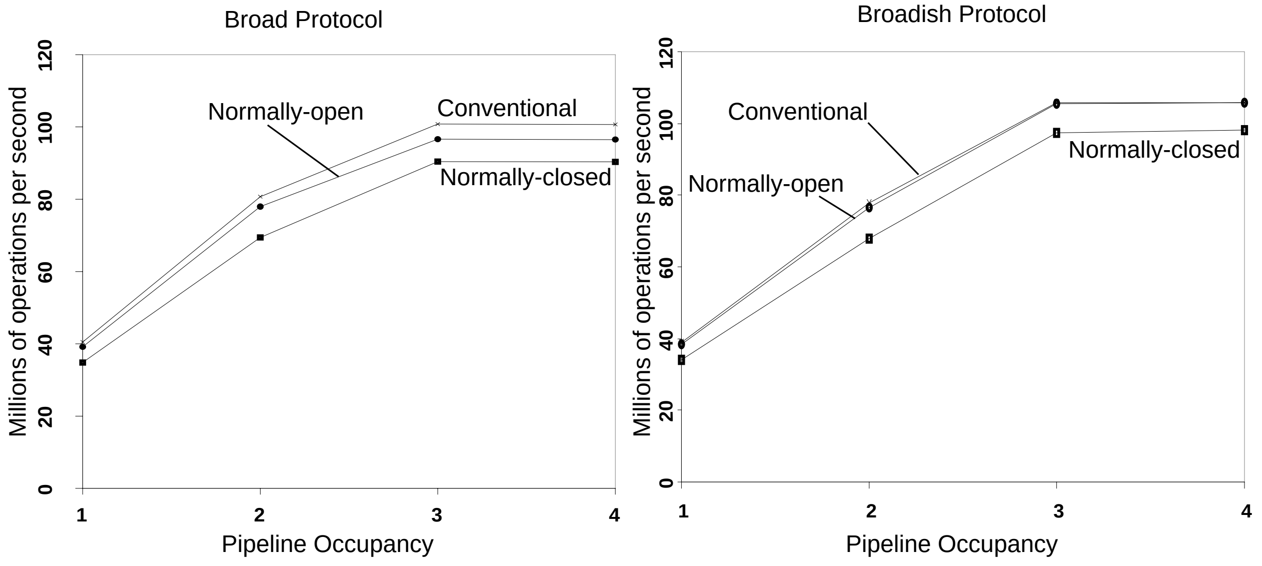

Figure 12. Plot of Throughput Against Pipeline Occupancy for Latch Controller Designs

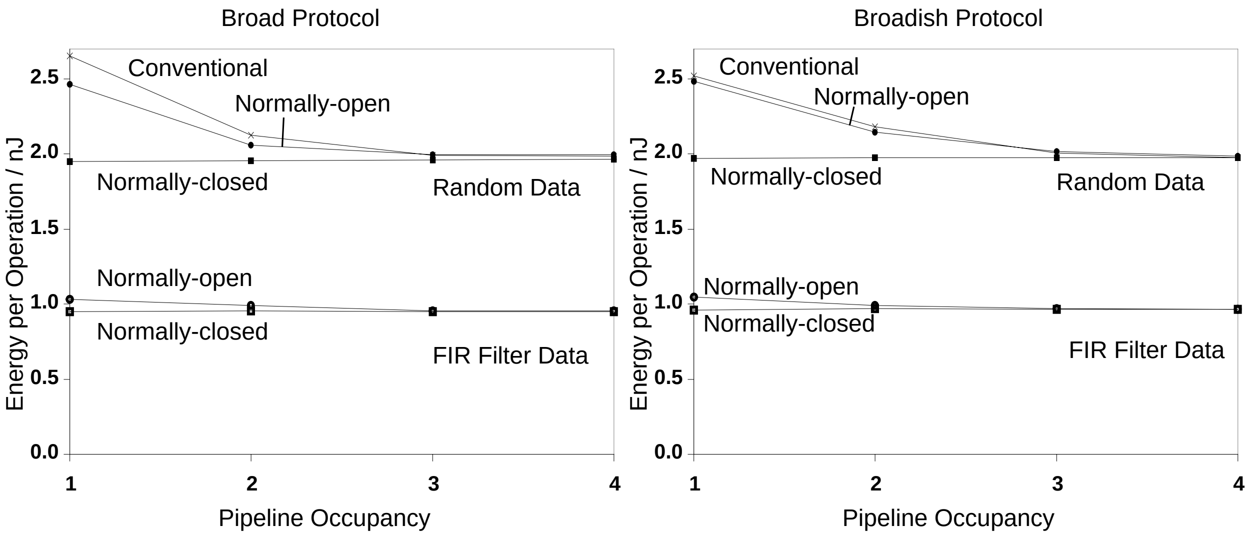

Figure 13. Energy per Operation Against Pipeline Occupancy for Latch Controller Designs

When using the broadish protocol configurable latch controller there is a negligible decrease in energy consumption in normally-closed mode as compared to normally-open mode at maximum throughput. The difference increases to 21% when operating with a single data item in the pipeline, and goes up to 32% when the multiplier and multiplicand inputs are also skewed in time.

8.3. Power Consumption Tests: FIR Filter Operation Data

Figure 13 shows the same trends when using the FIR filter data as seen with the random data. However, the average energy consumed per operation is around half that of the random data case. This is due to the correlations between successive data values, which means that fewer bits change on average than for random data. Also, the data input is held constant while each of the 8 filter coefficients is applied, leading to even less input activity. It should be noted that, due to simulation time constraints, only a few hundred sample values were processed. This means that there is no change in the characteristics of the signal during the period of simulation which may cause increased activity, such as a section of low-level signal or noise with frequent sign changes.

When using the broad and broadish protocol configurable latch controllers, an increase in energy per multiplication of only 7.7% and 8.3% respectively for the FIR filter data was observed between normally-open and normally-closed operation with a single data item in the pipeline. This is much less of a difference than was observed with random input data (21%), and again is due to the fact that the multiplicand input is held constant over each of the 8 FIR coefficients and that successive multiplicand values are correlated, leading to significantly less switching activity. Also, no increase in power consumption was observed when the inputs were skewed in time, as at least one input is always staying constant.

9. Conclusions

The results show that the use of reconfigurable latch controllers can make a significant difference to the power efficiency of an asynchronous system. The broadish protocol configurable latch controller in particular gives the possibility of reduced power consumption without a significant impact on maximum performance while the broad protocol configurable latch controller suffers a slight (4.2%) throughput penalty.

Improvement in overall power efficiency relies on there being a difference in the maximum and typical throughput required of the circuit. The best improvement in power efficiency is obtained when the pipeline is mostly empty during quiet periods. In the circuit under test, this would imply a throughput requirement 2-3 times greater in the maximum case than the typical case, which is a similar performance ratio to that of the Phillips DCC error correction circuit [2]. Assuming that similar differences in energy consumption could be obtained for this circuit as for the test circuit using random data, then reduction in average power consumption of between 16% and 24% could be expected at a greatly reduced circuit cost compared to the adaptive supply method.

The amount of power dissipated through spurious transitions is strongly dependent on the input signal characteristics. Where input operands to the circuit are skewed in time, the power saving in normally-closed mode is considerably greater than if they are presented simultaneously. However, in some instances where successive input values only change slightly, there appear to be fewer spurious transitions and so there is less benefit to be obtained from using the configurable-latch controllers.

The cost of the configurable controllers themselves is negligible. However, careful analysis is required of a given application to see whether or not the benefits obtained from using the configurable latch controllers outweigh the increased circuit cost of implementing the Turbo signal. If there is a significant difference between maximum and typical operating speeds, and the circuits are such that large numbers of spurious transitions are generated within the circuit, then the configurable latch controllers could make a very significant power saving.

10. Acknowledgements

This work formed part of the EPSRC/MoD Powerpack project, grant number GR/L27930. The authors wish to express their gratitude for this support.

References

[1] D. W. Dobberpuhl et al, “A 200Mhz 64-b Dual Issue CMOS Microprocessor”, IEEE Journal of Solid-State Circuits, Vol. 27, no. 11, pp 1555-1565, 1992

[2] L. S. Nielsen, C. Niessen, J. Sparsø,C. H. van Berkel, “Low power operation using self-timed circuits and adaptive scaling of the supply voltage”, IEEE Transactions on VLSI Systems, vol. 2, no. 4, pp 391-397, 1994.

[3] L. Nielsen, J. Sparsø, “A Low Power Datapath for a FIR Filter Bank”, Proc. International Workshop Symposium on Advanced Research in Asynchronous Circuits and Systems, IEEE Computer Society Press, March 1996

[4] I.E. Sutherland, “Micropipelines”, Communications of the ACM, vol. 32, no. 6, pp 720-738, June 1989.

[5] P. Day, J. V. Woods, “Investigations into micropipeline latch design styles”, IEEE Transactions on VLSI systems, vol. 3, no. 2, pp. 264-272, June 1995

[6] S. B. Furber, P. Day, “Four-phase micropipeline latch control circuits”, IEEE Transactions on VLSI systems, vol. 4, no. 2, pp 247-253, June 1996.

[7] S. Segars, “Low power microprocessor design”, MSc thesis, Department of Computer Science, University of Manchester, 1995.

[8] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, A. Yakovlev, “Petrify; a tool for manipulating concurrent specifications and synthesis of asynchronous controllers”, IEICE Transactions on Information and Systems, Vol. E80-D, No. 3, pp 315-325, March 1997.

[9] S. Furber, “A Small Compendium of 4-Phase Micropipeline Latch Control Circuits”, AMULET Group internal document, University of Manchester

[10] J. Liu, “Arithmetic and control components for an asynchronous microprocessor”, PhD thesis, Department of Computer Science, University of Manchester, 1997.