A TWO-PHASE MICROPIPELINE CONTROL WRAPPER FOR WITH EARLY EVALUATION

R. B. Reese, Mitchell A. Thornton, and Cherrice Traver

A two-phase control wrapper for a micropipeline is presented. The wrapper is implemented in an Artisan 0.13μ standard cell library that has not been augmented with any special cells for asynchronous design. The wrapper supports early evaluation allowing the output to be updated after a subset of the inputs have arrived, thus improving the throughput of the micropipeline.

Introduction: Micropipelines [1] use control logic wrapped around compute blocks to implement asynchronous systems. Micropipelines have been used to implement significant designs, including complex microprocessors [2]. Four-phase control [3] means that the control lines between micropipeline stages undergo a low-to-high-to-low transition for each data movement between stages; while two-phase control implies either a single low-to-high or high-low transition. Most micropipeline approaches use a bundled data signaling approach in which a single control wire is used for all data wires originating from a micropipeline stage. Delay elements are added to the control path to produce a matched control/datapath delay so that the latching signal from the control wrapper arrives at the output latches of the micropipeline stage at the same time as the data. Figure 1 shows the two-phase micropipeline control wrapper used in the design of a five-stage pipelined MIPS-compatible processor [5]. Each bundled data input *i* consists of a group of data lines *data_bundl_i* and its associated control line *Cin_i*. Each predecessor stage (fanin) provides a data bundle, and each successor stage (fanout) provides an acknowledgement signal. The control is two-phase, so each *Cin* input and acknowledgement

Submitted to *Electronics Letters*

will either all transition low-to-high, or high-to-low. After all *Cin* and acknowledgements have transitioned, then the C-element output transitions high-to-low or low-to-high. The XOR gate and *Cout* loopback signal generates a high-pulse on the GC signal when the C-element output changes state, latching the new outputs. The delay elements on the *Cin* inputs are used to match the delay of the control path to the compute function path.

A Two-Phase Wrapper with Early Evaluation: Figure 2 shows the wrapper of Figure 1 modified to support early evaluation. Early evaluation was used for performance enhancement of the microprocesor design presented in [5]. An *early fire* is defined as the *EE_sel* signal being a '1' after arrival of the early control inputs (the inputs to the *trigger* C-element). This causes the data (*Dout*) and control (*Cout)* signals to be updated after the *trigger* C-element toggles. The acknowledgement *Aout* is updated after all inputs have arrived, causing the *late* C-element to toggle. After an early fire, the input delays on the late-arriving inputs (the inputs to the late C-element) should be short circuited so that the acknowledgement (*Aou*t) is produced as quickly as possible once all inputs have arrived. Figure 3 shows the initial design of the *DKill* delay element. A single multiplexer cannot be used to bypass a long delay chain, because an input transition from the previous early fire may still be traversing the delay chain when the inputs for the next firing arrive, producing a hazard on the input to the late C-element. A *normal* fire occurs when *EE_sel* is a '0' after arrival of the early control inputs. In this case, the *Dout/Cout/Aout* outputs are updated after all control inputs arrive and the late C-element toggles. The delay block on the output of the *late* C-element is needed for a normal fire if the difference between the *Aout* delay and *Dout*/*Cout* delay paths is large, which can occur if the *GC* signal drives a large number of latch inputs. If *Aout* is provided too far in advance of *Dout/Cout*, a

predecessor block can change the input value to this stage, corrupting the compute function output value before it has been latched by the *GC* signal.

The asynchronous microprocessor design presented in [5] has been subsequently redesigned and synthesized to an Artisan 0.13μ standard cell library. C-elements wer mapped to standard cells using the approach in [4]. Pre-layout gate-level Verilog simulations using back-annotated SDF timing indicated the early evaluation wrapper design of Figure 2 was slow in producing an acknowledgment after an early fire occurred, primarily due to excessive loading on the late control input signals by the *DKill* block. The wrapper was also slow to produce a new *Dout* output when an early fire followed a normal fire (EE_sel '0' → '1') because of excessive loading by the *DKill* block on the *EE_sel* signal. Figure 4 shows a re-design of the early evaluation wrapper that has a dedicated C-element for producing a fast acknowledgement after an early fire. The *DKill* block was also redesigned as shown in Figure 5 to reduce loading on the *Cin* input signals and the *EE_sel* signal. The new *DKill* design uses two delay blocks; the toggling of the *sel* signal routes the *a* input between the two delay blocks so that one delay block is 'recovering' while the other delay block is 'active'. Normal operation is either a+ → N1+ (sel = 1) or a- → N0- (sel = 0) where the full delay chain penalty is used. An early fire can cause *sel* to change while the *a* transition is still within *dly1* or *dly0*. A change in *sel* chooses the opposite delay path, whose value is the normal arrival value for the previous delay path. The *Program Counter* block in the redesigned asynchronous microprocessor has six late inputs, and four early inputs; this block is used as an example in Table 1 to contrast the performance difference between the two wrapper designs. The maximum number of delay elements on a late control input was 9. Table 1 shows that the *Cin* to *Aout* delay after an early fire of the Version 2 wrapper is 34% less than the Version 1 wrapper. Neither wrapper used a delay block on the output of the late C-

element because of the low number of data outputs. The delay advantage of the Version 2 design would increase if usage of this delay block became necessary.

Conclusion: This paper introduces a two-phase control wrapper with early evaluation for a micropipeline block. The wrapper is intended for efficient mapping to a commercial standard cell library. The evolution of the wrapper design is traced through two different versions, with the second version containing an optimized path for acknowledgement output update after an early fire.

References

1. SUTHERLAND, I.. "Micropipelines", *Communications of the ACM*, Vol 32, No. 6, June 1989, pp. 720-738.
2. GARSIDE, J.D., FURBER, S.B., and CHUNG S.B.. "AMULET3 Revealed", In *Proceedings of the Fifth International Symposium on Advanced Research in Asynchronous Circuits and Systems*, Barcelona, Spain, April 1999, pp. 51-59.
3. FURBER, F.B., and DAY, P.. "Four-phase micropipeline latch control circuits", *IEEE Transactions on VLSI,* Vol. 4, No. 2, June 1996, pp. 11-16.
4. TOY-YUNG, W. and VRUDHULA, S. B.. "A Design of a Fast and Area Efficient Multi-Input Muller C-element", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol 1, No. 2, June 1993.
5. REESE, R.B., THORNTON, M.A., and TRAVER, C.. "A Coarse-grained Phased Logic CPU", In *Proceedings of the Ninth International Symposium on Advanced Research in Asynchronous Circuits and Systems*, Vancouver, BC, Canada, May 2003, pp 2-13.

Author's affiliations

Robert B. Reese (Department of Electrical and Computer Engineering, Mississippi State University, Box 9571, Mississippi State, MS 39762 USA) *reese@ece.msstate.edu*

Mitchell A. Thornton (Department of Computer Science and Engineering, Southern Methodist University, P.O. Box 750122, Dallas, TX 75275 USA) *mitch@engr.smu.edu*

Cherrice Traver (Electrical and Computer Engineering Department, Union College, Schenectady, NY 12308 USA) *traverc@union.edu*

List of Captions

Table 1: Performance Comparison

# of late Control inputs		6
Maximum # of Delay elements on late control inputs		9
# of Dout outputs		32
Cin to Aout Delay (ns)		%diff
Version 1	Version 2	
0.47	0.31	-34.0%

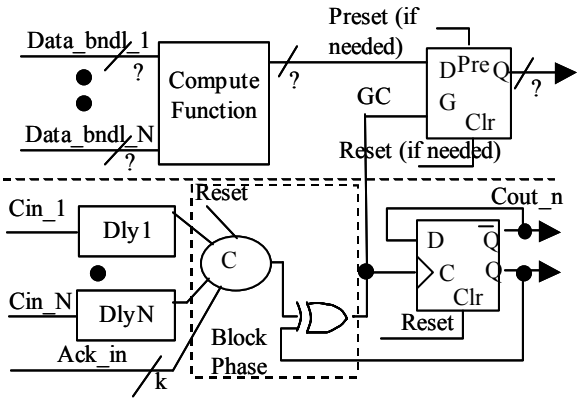

Figure 1: Micropipeline Wrapper for Two-Phase Control

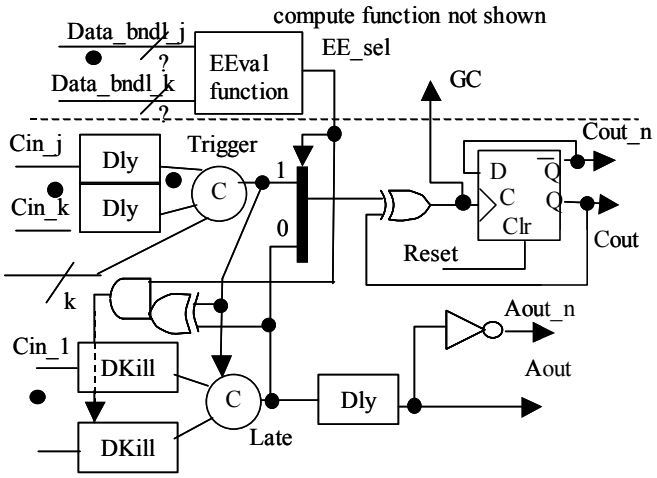

Figure 2: Micropipeline Wrapper with Early Evaluation (Version 1)

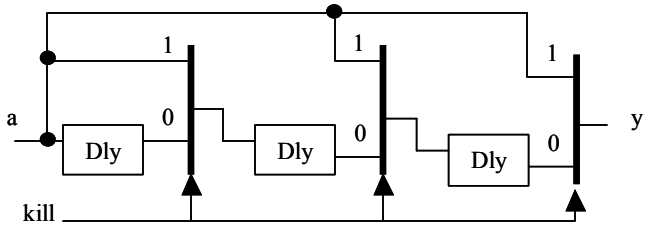

Figure 3: Delay Kill (Version 1, three delay stages shown)

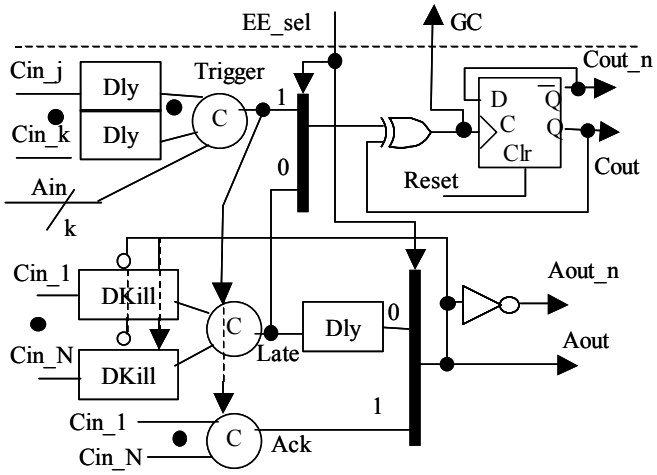

Figure 4: Micropipeline Wrapper with Early Evaluation (Version 2)

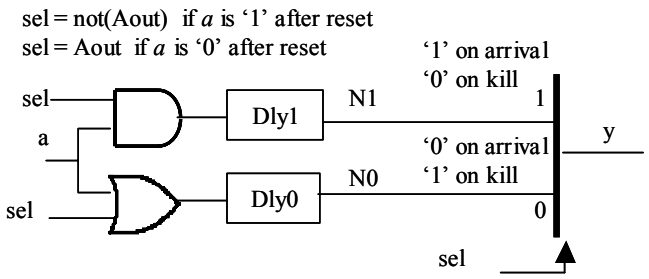

Figure 5: Delay Kill (Version 2)