Fast two-phase micropipeline control wrapper for standard cell implementation

R.B. Reese, M.A. Thornton and C. Traver

A fast control wrapper for a micropipeline with two-phase control is presented. The wrapper is implemented in an Artisan 0.13 μ standard cell library that has not been augmented with any special cells for asynchronous design. The wrapper is approximately 25% faster than a more traditional approach that uses a Muller C-element.

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. Typical micropipeline control logic use Muller C-elements which have efficient transistor level implementations for a small number of inputs (<4). Large input C-elements can be implemented as trees of smaller C-elements or can be mapped directly to standard cells as described in [4]. Most micropipeline approaches use a bundled data signalling 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. In designs with thin pipeline stages, the performance of the control logic becomes an issue, with the control path becoming the performance limiter instead of the datapath. Control logic performance is also important if a micropipeline stage has finished a computation, and is waiting on an acknowledgment from a successor stage to latch the new computation, thus providing the new value to the successor stage. Acknowledgments propagate backwards through the pipeline, and thus do not have delay elements in their path.

Fast two-phase wrapper: Fig. 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 C_{in_i}. Each predecessor stage (fanin) provides a data bundle, and each successor stage (fanout) provides an acknowledgment signal. The control is two-phase, so each C_{in} input and acknowledgment will either all transition low-to-high, or high-to-low. After all C_{in} and acknowledgments have transitioned, then the C-element output transitions high-to-low or low-to-high. The XOR gate and C_{out} 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 C_{in} inputs are used to match the delay of the control path to the compute function path. The delay element in the C_{out} loopback path to the XOR gate is used to control the pulsewidth of the *GC* signal for timeborrowing purposes. It should be noted that the C-element output could be used directly as the C_{out} signal; the use of the DFF in Fig. 3 for C_{out} allows easier control/datapath delay matching because both D_{out} and C_{out} are triggered by the same input signal, *GC*. A 0.13 μ standard cell library from Artisan was used to implement the processor presented in [5]. The C-element was mapped to standard cells using the approach in [4], as the Artisan standard cell library did not have an integrated C-element. Processor simulations using pre-layout, Verilog gate level simulations generated by the Synopsys synthesis tool indicated that the control logic path from C_{in} to D_{out} was the limiting performance factor in several blocks, either because the compute function delay was small, or because the block was triggered by arrival of an acknowledgment. The C-element and XOR gate was subsequently replaced by the logic shown in Fig. 2. This removed the XOR gate from the critical path of the control logic, and also reduced the delay of the arrival detection logic. The non-inverting delay in the multiplexer select path is used to increase the high pulsewidth of the *GC* signal. Table 1 contains performance results that compare the original implementation (Fig. 1) against the new wrapper logic (Fig. 2). The 'Control inputs' column is the total number of C_{in} and *Ack_in* inputs, while the 'Data inputs' column is the total number of data latches driven by the *GC* signal. Delays are in nanoseconds as reported by the Synopsys static timing analyser. No delay elements were used on the C_{in} inputs. The new wrapper has a significant performance advantage for control inputs up to 32, which would be an atypically large number of control inputs for a micropipeline stage that uses bundled data signalling. This performance advantage decreases as the number of control inputs becomes more than 32 (64, 256), which would only occur if the micropipeline was using a form of delay-insensitive dual-rail signalling between micropipeline stages. Fig. 3 gives the path detail for the original logic in the case of eight control inputs and 256 data outputs, while Fig. 4 gives the path detail for the new wrapper logic using the same test case. The standard cell naming convention is *gtype_k_X_n*, where k is the number of inputs for gate type *gtype*, and n is the drive strength. From Figs. 3 and 4, it is obvious that the new wrapper logic has a faster critical path, and that the XOR gate in the original design contributes a substantial portion to the total delay for this particular case.

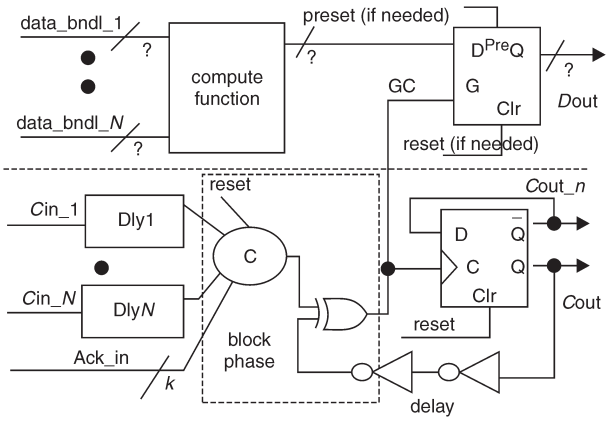

Fig. 1 *Micropipeline wrapper for two-phase control*

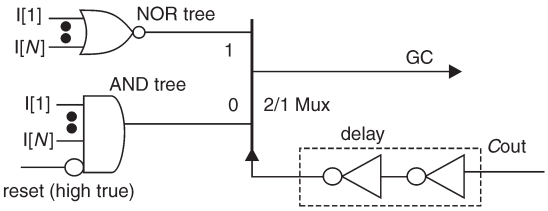

Fig. 2 *New arrival detection logic*

Table 1: Performance comparison

Control inputs	Data outputs	C_{in} to D_{out} delay (ns)		% diff
		Original	New	
4	16	0.47	0.31	−34.0
	64	0.49	0.35	−28.6
	256	0.54	0.42	−22.2
8	16	0.49	0.33	−32.7
	64	0.53	0.37	−30.2
	256	0.57	0.43	−24.6
16	16	0.51	0.41	−19.6
	64	0.54	0.45	−16.7
	256	0.59	0.51	−13.6
32	16	0.59	0.43	−27.1
	64	0.60	0.47	−21.7
	256	0.67	0.54	−19.4
64	16	0.60	0.58	−3.3
	64	0.63	0.62	−1.6
	256	0.68	0.69	1.5
256	16	0.69	0.62	−10.1
	64	0.72	0.65	−9.7
	256	0.77	0.72	−6.5
Average				−17.8

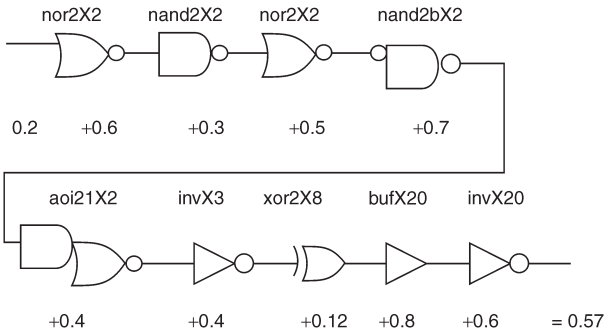

Fig. 3 *Path detail (original logic, eight control, 256 data)*

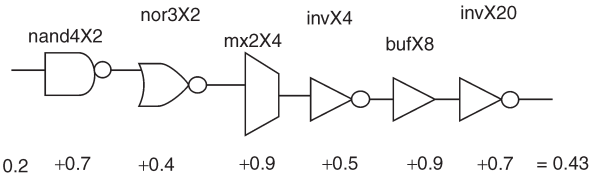

Fig. 4 *Path detail (new logic, eight control, 256 data)*

Conclusion: This Letter introduces a fast two-phase control wrapper for a micropipeline block. The wrapper is intended for efficient mapping to a commercial standard cell library that does not have specialised support cells such as C-elements for asynchronous design.

22 December 2003
Electronics Letters online no: 20040189
doi: 10.1049/el:20040189

R.B. Reese (*Department of Electrical and Computer Engineering, Mississippi State University, Box 9571, Mississippi State, MS 39762, USA*)

E-mail: reese@ece.msstate.edu

M.A. Thornton (*Department of Computer Science and Engineering, Southern Methodist University, P.O. Box 750122, Dallas, TX 75275, USA*)

C. Traver (*Electrical and Computer Engineering Department, Union College, Schenectady, NY 12308, USA*)

References

1 Sutherland, I.: 'Micropipelines', *Commun. ACM*, 1989, **32**, (6), pp. 720–738
2 Garside, J.D., Furber, S.B., and Chung, S.B.: 'AMULET3 Revealed'. Proc. 5th Int. Symp. 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 Trans. Very Large Scale Integr. (VLSI) Syst.*, 1996, **4**, (2), pp. 247–253
4 Toy-Yung, W., and Vrudhula, S.B.: 'A design of a fast and area efficient multi-input Muller C-element', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 1993, **1**, (2), pp. 215–219
5 Reese, R.B., Thornton, M.A., and Traver, C.: 'A coarse-grained phased logic CPU'. Proc. 9th Int. Symp. on Advanced Research in Asynchronous Circuits and Systems, Vancouver, BC, Canada, May 2003, pp. 2–13