Two-phase micropipeline control wrapper with early evaluation

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

A two-phase control wrapper for a micropipeline is presented. The wrapper is implemented in an Artisan 0.13 μm 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-to-low transition. 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 latch of the micropipeline stage at the same time as the data. Fig. 1 shows the two-phase micropipeline control wrapper used in the design of a five-stage pipelined MIPS-compatible processor [4]. Each bundled data input i consists of a group of data lines data_bndl_i and a single 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 be either all transition low-to-high, or high-to-low. After all C_in and acknowledgments have transitioned, then the C-element output transitions low-to-high 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_out inputs are used to match the delay of the control path to the compute function path.

Two-phase wrapper with early evaluation: Fig. 2 shows the wrapper of Fig. 1 modified to support early evaluation. Early evaluation was used for performance enhancement of the micropipeline processor design presented in [4]. 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 (D_out) and control (C_out) signals to be updated after the trigger C-element toggles. The arrival of all inputs causes the late C-element to toggle, which updates the acknowledgment (A_out) output. 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 acknowledgment (A_out) is produced as quickly as possible once all inputs have arrived. Fig. 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 D_out/C_out/A_out 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 A_out delay and D_out/C_out delay paths is large, which can occur if the GC signal drives a large number of latch inputs. If A_out is provided too far in advance of D_out/C_out, a predecessor block can change the input value to the stage, corrupting the compute function output value before it has been latched by the GC signal.

The asynchronous microprocessor design presented in [4] has been subsequently redesigned and synthesised to an Artisan 0.13 μm standard cell library. C-elements were mapped to standard cells using the approach in [5]. Pre-layout gate-level Verilog simulations using back-annotated SDF timing indicated that the early evaluation wrapper design of Fig. 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 $D_{out}$ output when an early fire followed a normal fire ($EE_{sel}$ '0' $\rightarrow$ '1') because of excessive loading by the $DKill$ block on the $EE_{sel}$ signal. Fig. 4 shows a redesign of the early evaluation wrapper that has a dedicated C-element for producing a fast acknowledgment after an early fire. The $DKill$ block was also redesigned as shown in Fig. 5 to reduce loading on the $C_{in}$ 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 $a+$ $\rightarrow$ $N1+$ ($sel=1$) or $a-$ $\rightarrow$ $N0-$ ($sel=0$) 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 $C_{in}$ to $A_{out}$ 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.

### Table 1: Delay comparison of wrappers

<table>
<thead>
<tr>
<th></th>
<th>Version 1</th>
<th>Version 2</th>
<th>%diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of late control inputs</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum number of delay elements on late control inputs</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of $D_{out}$ outputs</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{in}$ to $A_{out}$ delay (ns)</td>
<td>0.47</td>
<td>0.31</td>
<td>−34.0%</td>
</tr>
</tbody>
</table>

Conclusions: A two-phase control wrapper with early evaluation for a micropipeline block has been introduced. The wrapper is intended for efficient mapping to a commercial standard cell library. The evolution of the wrapper design has been traced through two different versions, with the second version containing an optimised path for an acknowledgment output update after an early fire.

© IEE 2004

Electronics Letters online no: 20040256
doi: 10.1049/el:20040256

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, PO Box 750122, Dallas, TX 75275, USA)

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

References