ECE 152B: Spring 2013 – Lab 2 Timing Handout

Nicole Lesperance
May 13, 2013

RTN Timesteps

The concrete RTN for each instruction consists of many timesteps, T_0 through T_N, where N is the number of clock cycles needed to complete the instruction. At each timestep, 2 things occur:

1. Control signals are set
2. Data is latched into state elements

Examples of control signals: IR_{in}, bus input selector, the M and S signals selecting the ALU operation, R/W, CE, OE signals for the SRAM, etc.

Examples of state elements: Registers $R0$ through $R3$, locations on the stack, the instruction register, the program counter, any flip flops on the inputs or output of the ALU, etc.

Control Signals

The control signals can be divided into 2 categories:

1. Gating signals (G): Control signals which affect combinational logic: bus control signals, M and S signals from the ALU, SRAM reading signals, etc.
2. State Element Input Signals (CS): Control signals which decide when to latch data into the state elements: IR_{in}, PC_{in}, SRAM writing signals, etc.

Timing Issues

The Goal: Figure out when to set both G and CS control signals properly for 1 RTN timestep. The main issue is that if G and CS signals are set simultaneously, there is no time for data to propagate before it is latched.

For example, think about the following RTN timestep which appears during ADD:

Step	RTN	Control Sequence
$T4$	$R[ra] \leftarrow A + R[rc]$	$R_{right,addr} = rc$, $R_{right,out}$, ALU_{add}, $R_{left,addr} = ra$, $R_{left,in}$, End

Assume the left port of the SRAM is used for writing data in, and the right port for reading data out.

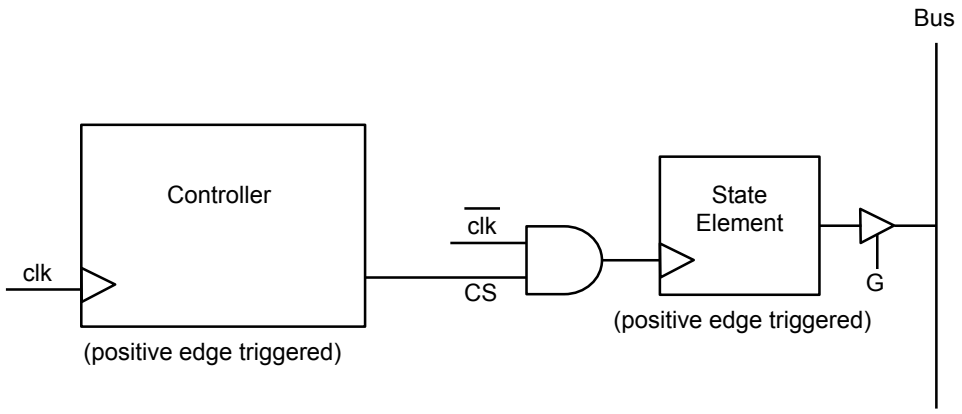
Data is being taken out of Rc, placed on Bus B, added with the value stored in A, then the result is written back into Ra all in the same clock cycle. The output of the ALU should not be written back into Ra until it is valid.

If $R_{left,in}$ controls when the data is latched into the SRAM, then this control signal must be set after all the other control signals to give enough time for the ALU to complete the add operation before the result is written back into Ra.

One Solution

There are many ways one can approach timing the control signals, but the following is a simple effective method:

The G signals are set on the positive clock edge at the beginning of the timestep and remain set the entire clock period. The CS signals are ANDed with an inverted clock, so that data is written into state elements on the negative clock edge, but the CS signals are set only during the negative half of the clock cycle.



In the timing diagram, the blue shaded region indicates the amount of time combinational signals have to propagate before data is latched into state elements. Since the data stays valid for the whole clock cycle, there is no worry about register hold times being violated (unless, of course, the clock period is too short).

In the example RTN above for the ADD instruction, the only CS signal is $R_{left,in}$. The rest of the signals are high throughout the entire clock cycle. In the block diagram, the State Element would be Ra being written to through the left port of the SRAM.

Another example: instruction fetch sequence.

Step	RTN	Control Sequence
$T0$	$IR \leftarrow PC$, $PC \leftarrow PC + 1$	IR_{in}, PC_{in}

For the instruction fetch, both control signals are CS signals, and would be set during the negative half of the clock cycle.