CSE140 DISCUSSION
WEEK 8 – HLSM, RTL

Jason Ma
Example RTL Problems

• You are tasked with something (HLSM word problem)
• You are given code which does something (C code to HLSM)

Approach

• Identify the inputs and outputs
• Identify what you may need to store locally (using registers)
• Draw HLSM
• Draw RTL and control unit
• Label connections between RTL with control units
RTL Datapath Components

• This is a great reference slide from lecture
• Highly recommend putting on cheat sheet if you have room
HLSM Word Problem

• Find instances where a string of bits matches a 4-bit pattern “0000”. The string of bits is at most 32 bits, and the output would be a corresponding binary string of 29 bits

• ie: A 1 in index 0 represents a match between indices 0-3 on the input string

• 00000100001 ...

• 11000010 ...

HLSM Word Problem

• Find instances where a string of bits matches a 4-bit pattern “0000”. The string of bits is at most 32 bits, and the output would be a corresponding binary string of 29 bits

• ie: A 1 in index 0 represents a match between indices 0-3 on the input string

• Identify the inputs and outputs:
  • Input: 32 bit string A (can represent as an array of bools)
  • Output: 29 bit string out (can represent as an array of bools)
HLSM Word Problem

• Find instances where a string of bits matches a 4-bit pattern “0000”. The string of bits is at most 32 bits, and the output would be a corresponding binary string of 29 bits

• ie: A 1 in index 0 represents a match between indices 0-3 on the input string

• Identify what you may need to store locally (using registers)
  • Output needs storage (out)
  • Could also store the pattern as an array of bools (P)
  • May need to also store the current index being compared (i)
HLSM Word Problem

• Let’s start drawing the HLSM:
  • We will need a for loop to iterate over the bit sequence, which can be implemented using while loop
HLSM Word Problem

• Inside the loop, we need logic to compare pattern and update output
• Why not do a single i++ before updating out? Does it make a diff?
HLSM Word Problem

• Let’s move on to the datapath
• The first thing we will need is the counter for i and registers to store our local values.
• NOTE: reg is supposed to be a 29 bit output, not 32
HLSM Word Problem

- The counter holds the address we care about.
HLSM Word Problem

• Next let’s look at how to slice the input.
• When i=0, we want bits 31-28
• When i=1, we want bits 30-27
• etc.
HLSM Word Problem

• Slicing continued:
HLSM Word Problem

• Comparing
HLSM Word Problem

- Add control unit
- Connect to datapath
- Is this complete?
HLSM Word Problem

- Still needs a few more things...
- Always check directionality
- Label multi-bit lines clearly
Timing

• Fill in timing diagram for following circuit with delay specifications:

• NOTE: NOT gate is actually supposed to be a 25ns buffer

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock frequency</td>
<td>5MHz</td>
</tr>
<tr>
<td>Flip flop $t_{ccq}, t_{pcq}$</td>
<td>50ns</td>
</tr>
<tr>
<td>Flip flop $t_{setup}, t_{hold}$</td>
<td>10ns</td>
</tr>
<tr>
<td>NOT gate $t_{cd}, t_{pd}$</td>
<td>25ns</td>
</tr>
<tr>
<td>OR gate $t_{cd}, t_{pd}$</td>
<td>50ns</td>
</tr>
<tr>
<td>AND gate $t_{cd}, t_{pd}$</td>
<td>50ns</td>
</tr>
</tbody>
</table>
Timing

• Clock period = 1/5MHz = 200ns
• Clock period is 4 bars
• One bar = 50ns
Timing

- NOT gate delay is exactly 25ns, or half bar
Timing

- NOT gate delay is exactly 25ns, or half bar
Timing

• May be useful to define and draw out S5 and S6.
Timing

• Then use S5, S2, and S6 to determine S3 and S4.
Timing

• S1, S2, S3, and S4 are captured as the following:
Questions

• Timing
• HLSM
• RTL