CIT 5930 Module05 HW: Datapath

Due Date: Wednesday 10/2 @11:59pm

Which Lectures Should I Watch to Complete this Assignment?: Module05 Video lectures can be found on canvas under “Class Recordings” PDF files of the module lectures can be found on canvas under “Files->Module_Slides”

What textbook sections should I read to complete this assignment? 4.1 We will begin to diverge a bit from the book at this point, but we are formally in chapter 4. Carefully read section 4.1 and the lecture slides for this week to help you with this HW.
Assigned Problems (to be c
From the textbook: 4.1, 4.2 xt of the “simple
processor” CPU presented
(Please consider using a sp IW!)

Custom Problem

a. For the NZP Tester (start are all the possible input
combinations for the NZP i they actually
mean.Example: $NZP = 0$
b. Implement an NZP Teste 16-bit # to test (from the
ALU), take in only a 3-bit # ation of the NZP tester.
Implement the “guts” of the

Custom Problem #2: Integer Division:

From the lecture slides, for the “Simple processor w/PC and NZP” shown on slide 5-37, implement the following division algorithm written in pseudocode below:

```
C = 0 ; // you don’t need to code this line
do { A = A – B ;
} while (A >= 0) ; C = C + 1 ;
C = C – 1 ;
```

NOTE: this algorithm uses a “do-while” loop. It is slightly different than a normal “while” loop. The computer will always go through the loop once, and then check the condition before it performs it again. This differs from a “while” loop that will only perform the loop IF the condition is first met.
a. Show a Flow Chart, like the one shown on slide 5-36, this will help you break the program into its

various “states.”
b. Show a table like the one shown on slide 5-41, listing the values of the control signals necessary to implement the algorithm above.
i. You may assign registers any way you like to hold your variables and necessary data, but state it up front & give their initial values.
ii. You don’t need to have instructions to set values of the registers.
c. Show an “Execution Trace” like the one shown on slide 5-43 for your program in step (b). For your trace,assume $A = 5$ and $B = 2$

Custom Problem #3: Summing Up Numbers:

For this problem you will once again use the “Simple processor w/PC and NZP” shown on slide 5-37. This time you will create the algorithm that you will implement. Your algorithm will take as input an unsigned number: “A”. It m [illegible] om 1 up to but not including A. As an example [illegible] ng: $1 + 2 + 3 + 4$ You may assume (as we have in [illegible] dy loaded in your register file, as well as any other nu [illegible] ement your algorithm.

a. Write out your algorithm [illegible] #3 for your algorithm.
b. Show a flowchart of your [illegible]
c. Show a table like the one [illegible] ontrol signals necessary to implement your algorithm. [illegible] n them, and you can assign the register to hold a [illegible]
d. Show an “Execution Trac [illegible] ogram in step (c). For your trace,assume $A = 3$

Custom Problem #4: Timing in the CPU:

For the “execution trace” shown on slide 5-43 of the datapath lecture, create a timing diagram for all 12 signals shown in the table: PC, +/-, AR1, etc. For all signals wider than 1 bit, you are welcome to use the abbreviated form of the multi-bit signal shown in module04 (sequential logic) slide 4-46. The next page of this assignment has the timing diagram that you must complete on it.
You need only to show the first 3 cycles of the clock from the execution trace

- These specifications may or may not be useful in the construction of the timing diagram
- The period of the clock is 10ns
- The propagation delay for reading from the register file is: 2.5ns
- The propagation delay for writing to the register file is: 3.5ns
- The propagation delay for reading from the control memory is: 1ns

- The propagation delay for the ALU is 2ns for addition
- The propagation delay for the ALU is 2.5ns for subtraction
- The propagation delay for the Test box is 0.5ns
- The propagation delay for Test MUX is 0.5ns
- The propagation delay for the incrementor box is 1.5ns
- The propagation delay for the PC register is 1.5ns o notice on the timing diagram that it took 1. ns for the $PC = 0$ after the clock went from 0 to 1 – you must account for the propagation delay of each component like this, e.g. – the ALU takes 2ns for addition

After constructing the timing diagram, answer the following:
a) Is the clock period long enough? If not, what should it be?
b) If the clock period is long enough, how short could you make it and still have the CPU function properly?
c) What is the frequency of

Directions on how to submi omework with
pencil/paper first and keep signment (using your
favorite app) and upload it a nvas (as you did for the last
HW).# cit593 cit5930 HW D