

ECE 368, Spring 2006, Instructor: Prof. Shantanu Dutt

Final Exam: Wed., May 4, Time: 1-3 pm

Exam Format: Open book and notes except for lab submissions

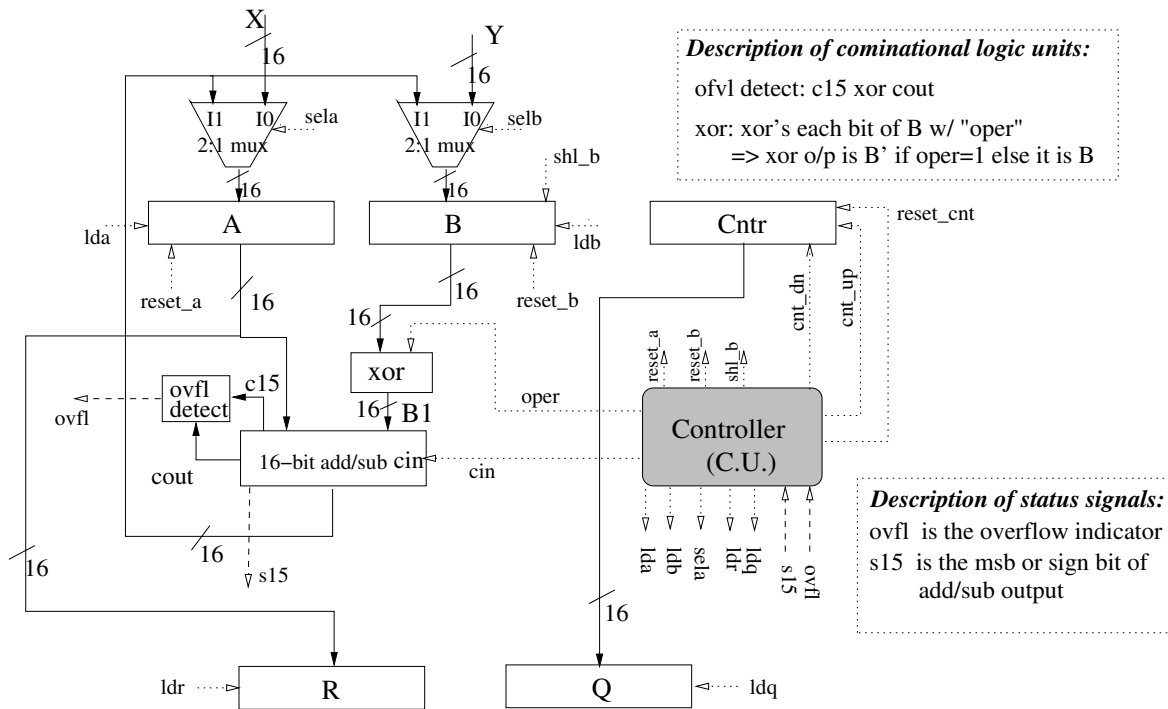
Total Points: 100

Important Note: (1) Write your name on the top of this question sheet and submit along with your answer book

(2) You need to show all your work **clearly** in deriving the answers. Just writing down the final answers is not enough.

Suggestion: Begin by reading all questions and do those first that you think you know best.

1. Digital Design – Datapath Control



Description of control signals:	
sela = 0 (1) means X (add/sub o/p) is conn. to input of reg. A	oper = 0 (1) means B1 = B (B')
lda = 1 means load/write A	cnt_up = 1 means Cntr \leftarrow Cntr + 1
reset_a = 1 means A \leftarrow 0	cnt_dn = 1 means Cntr \leftarrow Cntr - 1
ldb = 1 means load/write B	reset_cnt = 1 means Cntr \leftarrow 0
reset_b = 1 means B \leftarrow 0	ldr = 1 means load/write R
shl_b = 1 means contents of B are shifted left by 1 bit	ldq = 1 means load/write Q
cin is the carry input to add/sub unit	

NOTE: All write/cnt up operations complete only at the +ve edge of the next cc (since they all load new values into regs)
 NOTE: Signal lines without any specified width are 1-bit wide

(a) For the above datapath, design the Moore state diagram of the FSM for the control unit that executes the following function in the datapath described below using regular programming language statements:

```

begin
cntr=0; A=X; B=1; /* X is a positive integer of max value 214 - 1 */
while (A ≥ B) do begin
B = 4*B; cntr=cntr+1;
endwhile
cntr=cntr-1; Q=cntr;
end

```

IMPORTANT NOTES:

- Not all control signals may be needed for executing the above function.
- Y can have any arbitrary value. It is not necessarily 1.
- Using the datapath, you need to determine a way to get the integer value 1 in the B register before beginning to execute the **while** loop.
- Assume that all operations in the datapath except add/sub takes 1 cc, and add/sub takes 2 cc's.
- State beside each state of your FSM the RTL description of what is being accomplished in that state.

You will be graded most importantly on the correctness of your FSM controller, and then on the speed and number of states of your design. **40**

(b) What function of X is stored at the end in Q? Illustrate with an example. **7**

(c) Determine the delay in cc's and as a function of X , that your FSM controller takes to execute the above function. **8**

2. VHDL Coding & Module Communicatiuon

A module (a piece of hardware that does something useful) “something” is to be described in VHDL with the following specifications. It has two bit inputs $start1$, $start2$ and two integer inputs x , y . Inputs $start1$ and x are produced by one external module and inputs $start2$ and y are produced by another external module. Each bit inputs are asserted (made = '1') for exactly 5 ns by the corresponding external module. $start1$ becomes '1' when input data x is available and $start2$ becomes '1' when input data y is available. The input data (x , y) are integer values and are also available for exactly 5 ns each along with their respective signals $start1$, $start2$. There is no synchrony between the availability of data on the x and y inputs. Thus note in particular that $start1$ and $start2$ may or may not become '1' simultaneously, and in the case that are not '1' simultaneously, it is not known in which order they will become '1'.

Module “something” is supposed to produce the real output $z = x^2 + y^2$ with a delay of 50 ns **after both input bit signals $start1$, $start2$ have become '1' in the current iteration. At the same time** that z is produced, module “something” also asserts an output bit signal $done$ and outputs z for exactly 5ns.

Note also that *unresponsive signalling* is used between the modules, meaning they are busy waiting for the others' signal(s).

Give the entity and behavioral architecture description of module “something”. 25

3. Miscellaneous

(a) How are signals useful in VHDL? 5

(b) How are variables useful in VHDL? 5

(c) Mention and briefly discuss two major augmentations available in VHDL to describe computations/functions (that will finally be performed in hardware), that are not available in regular sequential programming languages like C/C++. 10