CS226 End-Semester Examination (Spring 2017)

Max marks: 60 **Time: 180 mins**

- ***The exam is open book and notes brought to the exam hall.***
- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- *If you need to make any assumptions, state them clearly.*
- ***Please start writing your answer to each sub-question on a fresh page. DO NOT write answers to multiple sub-questions on the same page.***
- ***The use of internet enabled devices is strictly prohibited. You will be debarred from taking the examination if you are found accessing the internet during the examination. All IIT Bombay rules apply in this respect.***
- ***Please do not engage in unfair or dishonest practices during the examination. Anybody found indulging in such practices will be referred to the D-ADAC.***

1. *[20 marks]* Consider the datapath shown in Fig. 1, in which all operations are on signed integers. This datapath consists of one adder, one adder/subtractor unit, and one multiplier+comparator unit, 8 registers labeled $R_1, \ldots R_8$, all of whom have the same bit-width, and 8 multiplexors, some of which share common control inputs (see, for example, $M3$ and $M4$).

   The input AS of the adder/subtractor unit controls whether the unit adds (AS = 1) or subtracts (AS = 0). The output of the multiplier+comparator unit labeled "A¡B" is set to 1 if and only if the input A is *strictly* less than the input B, both being treated as signed integers. It is assumed that the bit-width of all registers and data connections (bold black arrows) are the same, and this is assumed to be wide enough for all required calculations (i.e. there are no overflows).

   We wish to implement a controller that implements the following algorithm on this datapath. Please fill in the controller table in the template shown below for this purpose. All registers are assumed to have synchronous resets. In the interest of cautious design, you *must not* try to reset a register and have the data prepared at its inputs in the same cycle (recall discussion in class).

   You may assume that the inputs In1 and In2 are not changed until the entire computation is over, and done is set to 1.

```
done := 0;
sum := 0;
read In1;
read In2;
diff := In2;

while (sum < diff) {
   sum := sum + In1;
   diff := diff - 1;
}

result := sum * diff;
done := 1;
```

Fill in the controller table completely and accurately, so that the algorithm given above is implemented on the datapath. Assume AS is fixed to 0.

CurrState	$A < B$	NxtState	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$	Reset	Done	Comment
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\vdots$											

2. *[15 marks]* You are given two sequential circuits, C_1 and C_2, that are connected through a combinational circuit D, as shown in Fig. 2.

   All flip-flops in both C_1 and C_2 have the following characterisitcs:
   (i) Clk-to-Q delay: 10ps, (ii) Setup time: 7ps, (iii) Hold time: 11ps.

   The clock skew in circuit C_1 is 15 ps, and that in circuit C_2 is 20 ps.

   The maximum clock speed at which circuit C_1 can run independently (when not connected to C_2 through D) is known to be 2 GHz. Similarly, the maximum clock speed at which circuit C_2 can be run independently is known to be 3.3 GHz (1 GHz = 10^9 Hz).

   We want circuits C_1 and C_2 must run at the maximum permissible common clock frequency when they are connected to each other through D as shown in Fig. 2. Find the maximum such permissible common clock frequency.

   You are told that no combinational loops (or cycles) are formed when C_1 and C_2 are connected as in Fig. 2. In addition, you are told that the minimum and maximum delay from any input of D to any of its outputs are 100 ps and 20 ps respectively.

3. *[10+15 marks]* You are given a circuit representing a Boolean function $\varphi(y, x_1, \ldots x_n)$ of $n+1$ variables. You are required to implement an n-input Boolean function $\psi(x_1, \ldots x_n)$ with only $x_1, \ldots x_n$ as inputs such that the following condition holds:

   - $\exists y,\ \varphi(y, x_1, \ldots x_n) \equiv \varphi(\psi(x_1, \ldots x_n), x_1, \ldots x_n)$.

   In other words, for every value of $x_1, \ldots x_n \in \{0, 1\}^n$, if there are one or more values of y that make φ true, then $\psi(x_1, \ldots x_n)$ must be one such value. Of course, if there are no values of y that make φ true for the given values of $x_1, \ldots x_n$, then we don't care about the value of ψ.

   - Show that both $\varphi(1, x_1, \ldots x_n)$ and $\neg\varphi(0, x_1, \ldots x_n)$ serve as the function ψ required above.
   - In general, there can be multiple functions ψ that satisfy the requirements above for a given φ (we just saw two such ψ's in the previous sub-question.

     Show that there exists a function $\alpha(x_1, \ldots x_n, z)$, where z is distinct from y (i.e. z is a fresh variable) such that the following hold:

     - For every Boolean function $\beta(x_1, \ldots x_n)$, the function $\alpha(x_1, \ldots x_n, \beta(x_1, \ldots x_n))$ satisfies the requirements for ψ stated in the question.
     - Every ψ that satisfies the requirements stated in the question can be obtained by substituting some Boolean function $\beta(x_1, \ldots x_n)$ for z in α.

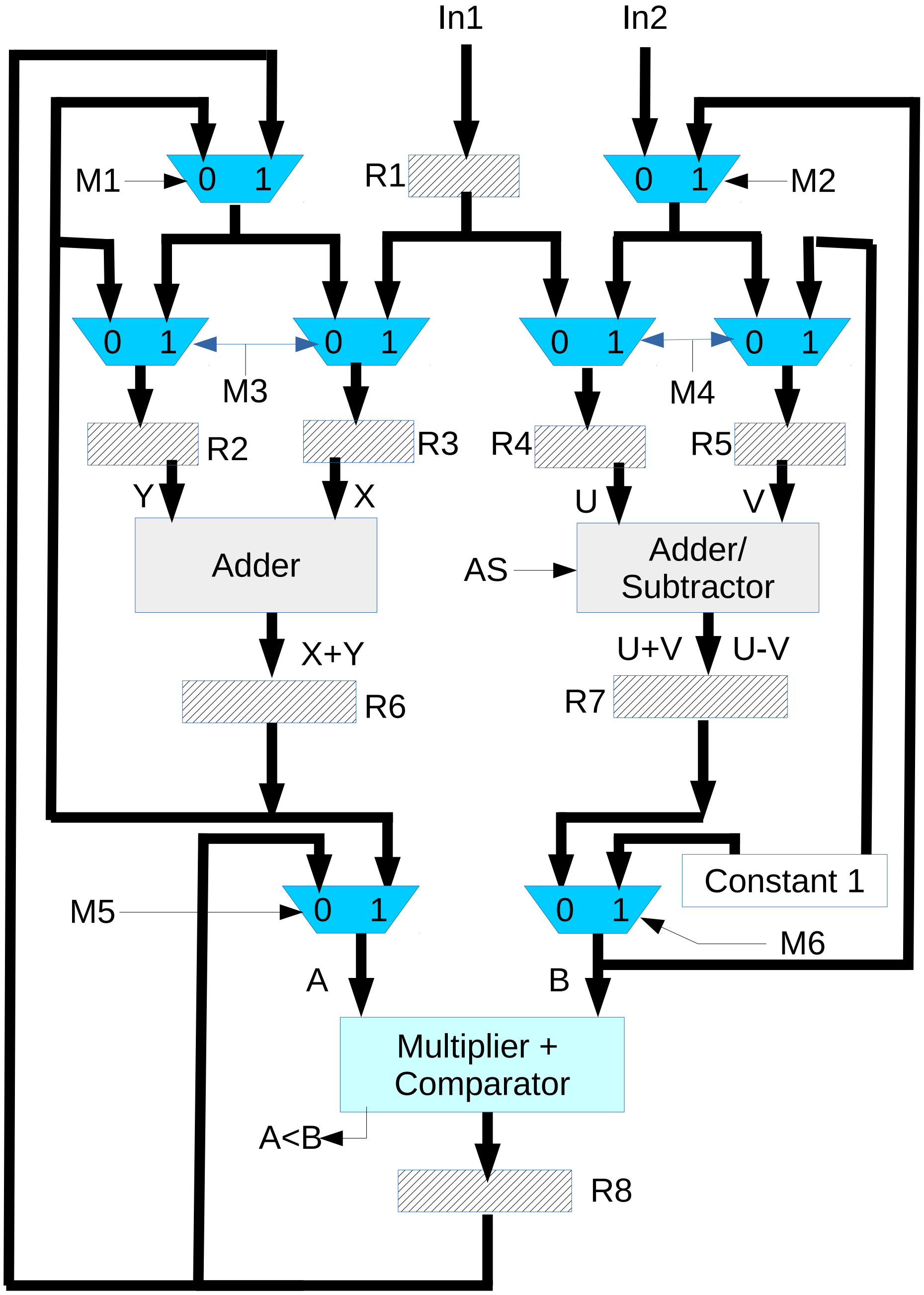

Figure 1: Datapath for Question 1

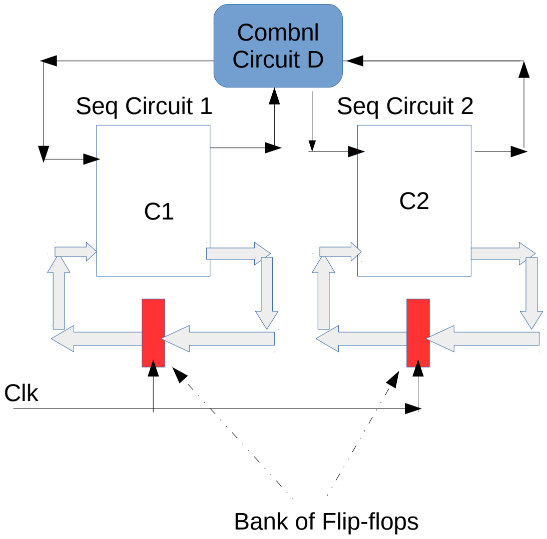

Figure 2: Circuits for Question 2