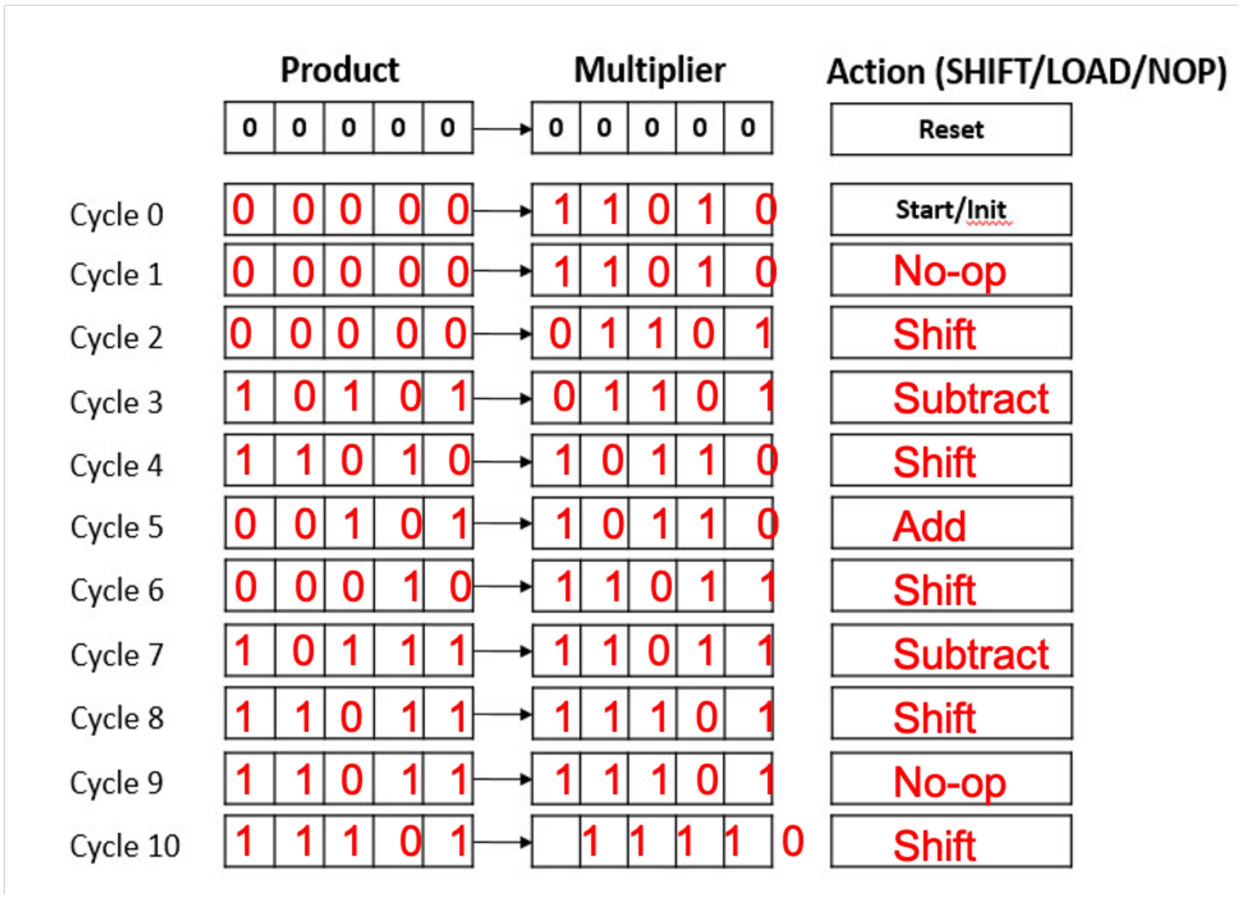
CPEG 422/622 Spring 2020

Homework 4

Due April 3rd at midnight (through Canvas)

1. Suppose there is a 5-bit Booth's multiplier, the multiplicand is **01011** and multiplier is **11010**, all in two's complement form. Please fill in the snapshot value of *partial product* register and *multiplier* register at each cycle, as well as the corresponding operation in the cycle.



	Product	Multiplier	Action (SHIFT/LOAD/NOP)
	0 0 0 0 0	0 0 0 0 0	Reset
Cycle 0	0 0 0 0 0	1 1 0 1 0	Start/Init
Cycle 1	0 0 0 0 0	1 1 0 1 0	No-op
Cycle 2	0 0 0 0 0	0 1 1 0 1	Shift
Cycle 3	1 0 1 0 1	0 1 1 0 1	Subtract
Cycle 4	1 1 0 1 0	1 0 1 1 0	Shift
Cycle 5	0 0 1 0 1	1 0 1 1 0	Add
Cycle 6	0 0 0 1 0	1 1 0 1 1	Shift
Cycle 7	1 0 1 1 1	1 1 0 1 1	Subtract
Cycle 8	1 1 0 1 1	1 1 1 0 1	Shift
Cycle 9	1 1 0 1 1	1 1 1 0 1	No-op
Cycle 10	1 1 1 0 1	1 1 1 1 0	Shift

The final result is 11101 11110 (which is -66).

+10 pts for the correct sequence of action: no-op, add, sub, no-op.
+10 pts for the alternating action/shift sequence.
+20 pts for the correct final result.

2. For the following examples of signals and variables, report the value of RESULT when the TRIGGER signal changes. Each table shows the waveform of TRIGGER. Fill in each table with the value of RESULT. Briefly explain your answer.

A process using variables

```
architecture VAR of EXAMPLE is
        signal TRIGGER, RESULT: integer := 0;
begin
        process
                variable var1: integer :=1;
                variable var2: integer :=2;
                variable var3: integer :=3;
        begin
                wait on TRIGGER;
                var1 := var2;
                var2 := var1 + var3;
                var3 := var2;
                RESULT <= var1 + var2 + var3;
        end process;
end VAR
```

A process using signals

```
architecture SIGN of EXAMPLE is
signal TRIGGER, RESULT: integer := 0;
signal sig1: integer :=1;
signal sig2: integer :=2;
signal sig3: integer :=3;
begin
process
begin
wait on TRIGGER;
sig1 <= sig2;
sig2 <= sig1 + sig3;
sig3 <= sig2;
RESULT <= sig1 + sig2 + sig3;
end process;
end SIGN;
```

The variable case:

	0 ns	5 ns	10 ns	15 ns	20 ns	25 ns	30 ns	35 ns
TRIGGER	1	0	1	0	1	0	1	0
var1	2	5	10	20	40	80	160	320
var2	5	10	20	40	80	160	320	640
var3	5	10	20	40	80	160	320	640
RESULT (after exiting process)	12	25	50	100	200	400	800	1600

var1, var2 and var3 are computed sequentially; their values are updated instantaneously after the TRIGGER signal arrives. Thus in the first iteration we have: var1 = 2; var2 = 2+3 = 5; var3 = 5;
RESULT, which is a signal, is computed using the new values of the variables. It is updated when exiting the process. Thus in the first iteration we have RESULT = 2 + 5 + 5 = 12; -- after exiting the process
The other iterations follow the same rules.

+ 3 pts for each value of the result (24 pts total)
+ 3 pts for explaining that variables are updated instantaneously.
+ 3 pts for explaining that RESULT is updated when exiting the process.

The signal case:

	0 ns	5 ns	10 ns	15 ns	20 ns	25 ns	30 ns	35 ns
TRIGGER	1	0	1	0	1	0	1	0
sig1 (inside process)	1	2	4	4	8	8	16	16
sig2 (inside process)	2	4	4	8	8	16	16	32
sig3 (inside process)	3	2	4	4	8	8	16	16
RESULT (after exiting process)	6	8	12	16	24	32	48	64

sig1, sig2 and sig3 are also computed sequentially; their values are not updated until exiting the process.
Thus in the first iteration we have:
sig1 = 2; sig2 = 1+3 = 4; sig3 = 2; -- after exiting the process (shown at 5ns column)
RESULT, which is a signal, is computed using the current values of the signals. It is updated when exiting the process. Thus in the first iteration we have RESULT = 1 + 2 + 3 = 6; -- after exiting the process
The other iterations follow the same rules.

+ 3 pts for each value of the result (24 pts total)
+ 3 pts for explaining that signals are updated when exiting the process.
+ 3 pts for explaining that RESULT is updated when exiting the process.