CPE 233: Software assignment 3

Prof. Bridget Benson

Luis Gomez, Jared Rocha

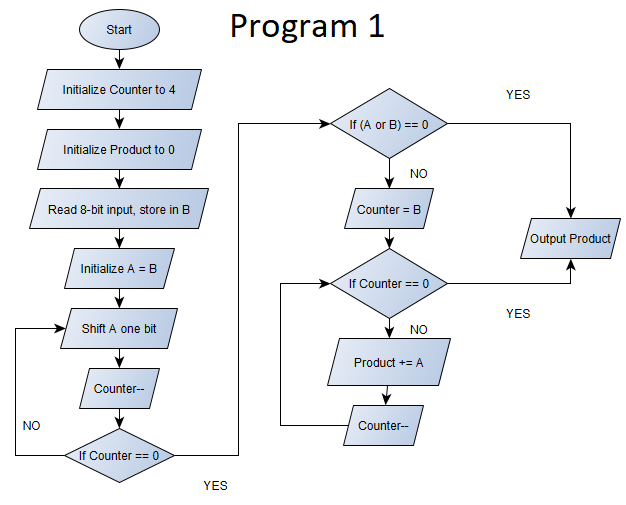
# Behavior

In this assignment, we wrote two Assembly programs using the RAT simulator.

**Program 1**: A multiplier program that reads a single 8-bit value, and outputs the product of A x B; where A is composed of the 4 most significant bits and B by the 4 least significant bits. Value B is generated by applying a mask to the 8-bit input to isolate the needed bits. Value A is generated by right-shifting the 8-bit value four times, clearing the carry with each iteration. When A & B are isolated, we check for zero and if not zero, we computer the product by adding A to a running total B times. The resulting product it output.

Program 2:

# Flowchart



Program 2

# Verification

# Source Code

TESTBENCH

**module PC\_and\_PCMUX\_tb**(); // Normally we would define input/output inside here, instead our inputs/outputs are internal logic signals found below

// clock signal

logic CLK;

// PC\_MUX interface

logic [9:0] FROM\_IMMED, FROM\_STACK;

logic [1:0] PC\_MUX\_SEL;

// Program Counter interface

logic PC\_LD, PC\_INC, RST;

logic [9:0] PC\_COUNT;

logic [9:0] DIN;

PC\_PCMUX\_interface DUT\_interface(.\*);

always // Sim Clock signal

**begin**

CLK = 0; #50; // 5 nanosecs

CLK = 1; #50;

**end**

initial **begin**

// Test 1: NO LOAD, DIN = FROM\_IMMED

FROM\_IMMED = 10'hA; FROM\_STACK = 10'hB;

PC\_MUX\_SEL = 0; PC\_LD = 0; PC\_INC = 0; RST = 0;

#100; // Test 2: : LOAD, DIN = FROM\_STACK

PC\_LD = 1; PC\_MUX\_SEL = 1;

#100; // Test 3: RESET

RST = 1; PC\_LD = 0;

#100; // Test 4: : INCREMENT

PC\_INC = 1; RST = 0;

#200;

$finish;

**end**

**endmodule**

PC MUX INTERFACE

**module PC\_PCMUX\_interface**(

// clock signal

**input** CLK,

// PC\_MUX interface

input [9:0] FROM\_IMMED, FROM\_STACK,

input [1:0] PC\_MUX\_SEL,

// output logic [9:0] DIN

// Program Counter interface

input PC\_LD, PC\_INC, RST,

output logic [9:0] PC\_COUNT

// output logic [9:0] DIN // ENABLE DURING TEST BENCH TO SEE SIGNAL

)**;**

logic [9:0] DIN;

pc\_MUX PCMUX(FROM\_IMMED, FROM\_STACK, PC\_MUX\_SEL, DIN);

program\_counter PC(DIN, PC\_LD, PC\_INC, RST, CLK, PC\_COUNT);

endmodule