ECE2330 – Digital Logic Design

Instruction Sequencer FSM

Learning Objective

After this activity, you will be able to

- Design circuit to generate the instruction sequencing signals using analytical digital design techniques, and
- Verify that the implementation you specify satisfies the requirements

Problem Statement

For this learning activity, you are **required** to design a circuit to generate the instruction sequencing signals used to ensure the proper sequencing of the control signals generation for instruction fetch and execute, which will effectively complete your control unit design.

The circuit is a Finite State Machine (FSM) with 3 inputs (that is, the 3-bit opcode) and requires 3 sequential elements to store the current state (because the maximum number of steps to execute the Add, Sub, Inc, and Dec instructions is 8). A state transition table for the circuit is shown in Figure 1 (but note that the table is not organized according to the Gray code). An 'x' indicates a "don't care" because under normal operation that situation should not occur. Also, a state transition diagram is shown in Figure 2, where **&&** is a shorthand notation for logical AND (|| – two vertical bars – is a shorthand notation for logical OR).

Inputs: Opcode / Current State	Next State: Load (000)	Store (001)	Add (010)	Sub (011)	Inc (100)	Dec (101)	Bra (110)	Beq (111)
000	001	001	001	001	001	001	001	001
001	010	010	010	010	010	010	010	010
010	011	011	011	011	011	011	011	011
011	100	100	100	100	100	100	100	100
100	101	101	101	101	101	101	000	000
101	000	000	110	110	110	110	x	x
110	x	x	111	111	111	111	x	x
111	x	x	000	000	000	000	x	x

Figure 1: Simple CPU Sequencer FSM State Transition Table

The function of this circuit is to generate the 8 timing pulses that are used to sequence the instruction fetch and execute operations. A typical set of outputs is shown in Figure 3, for an instruction that requires 8 steps to complete (for example, an **Add** instruction). The output table for the timing pulses is shown in Figure 4 for 8 instructions.

The circuit is also required to generate a signal to indicate when an instruction is being fetched and when it is begin executed. During timing pulses T0-T3, **Execute** = 0, indicating the instruction is being fetched. Conversely, for the remaining timing pulses for a given instruction, **Execute** = 1, indicating the execution of the current instruction. The output table for Execute is shown in Figure 5.

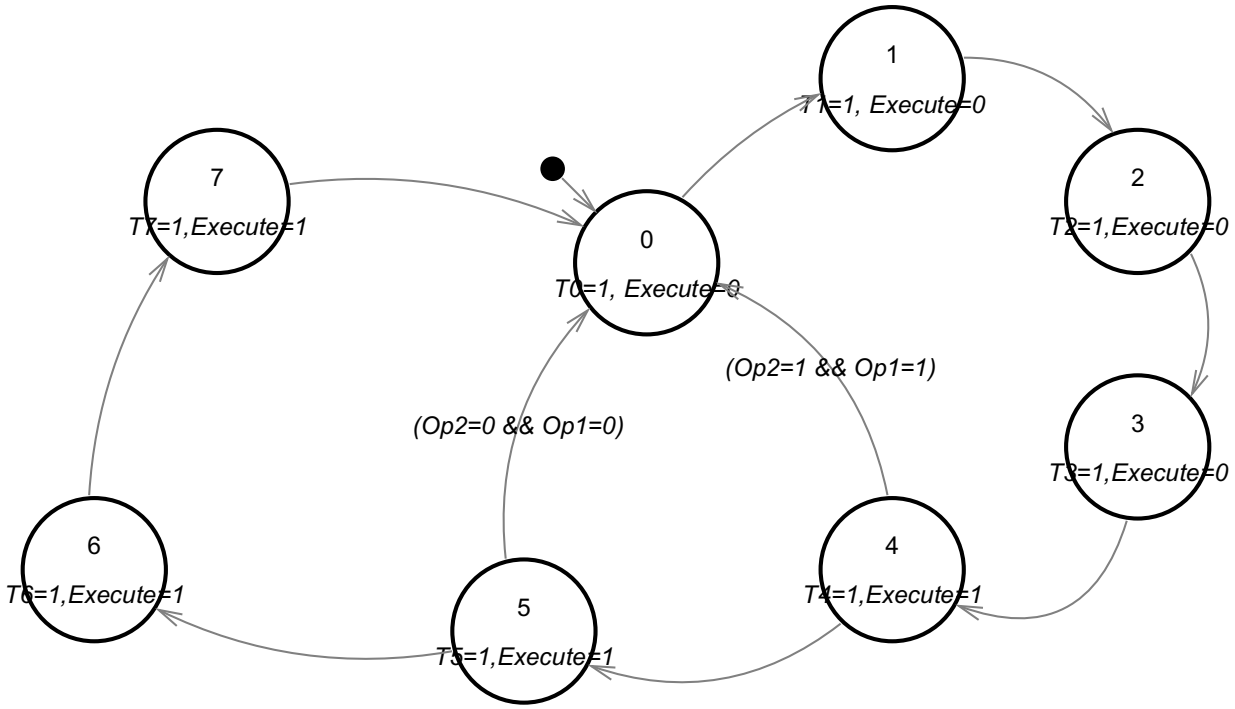

Figure 2: FSM State Transition Diagram

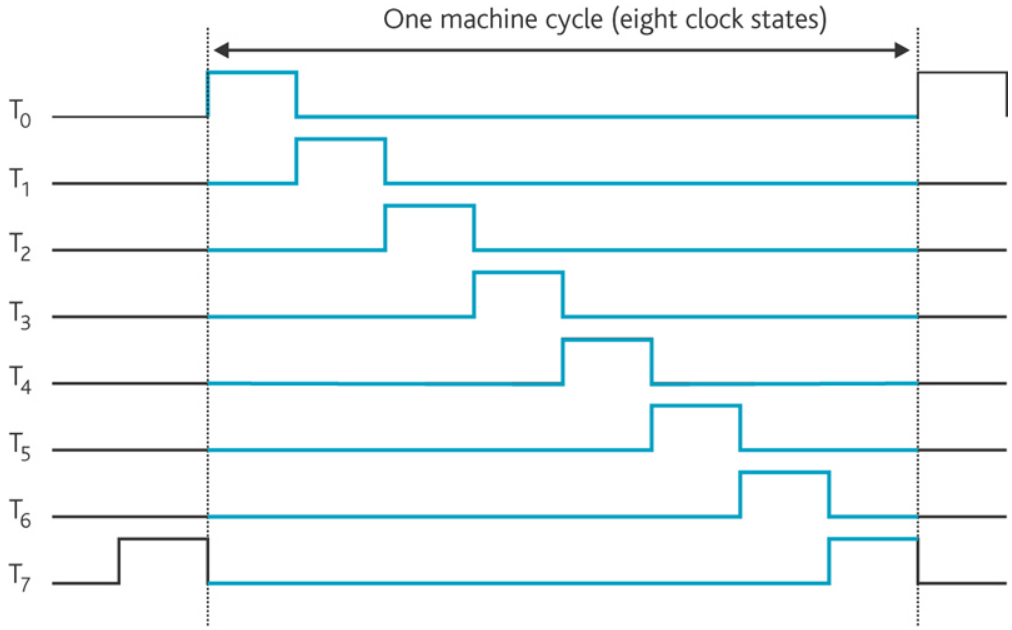

Figure 3: Timing pulses for one instruction cycle.

Output: T0-T7

Inputs: Opcode / Current State	Load (000)	Store (001)	Add (010)	Sub (011)	Inc (100)	Dec (101)	Bra (110)	Beq (111)
000	T0	T0	T0	T0	T0	T0	T0	T0
001	T1	T1	T1	T1	T1	T1	T1	T1
010	T2	T2	T2	T2	T2	T2	T2	T2
011	T3	T3	T3	T3	T3	T3	T3	T3
100	T4	T4	T4	T4	T4	T4	T4	T4
101	T5	T5	T5	T5	T5	T5	x	x
110	x	x	T6	T6	T6	T6	x	x
111	x	x	T7	T7	T7	T7	x	x

Figure 4: Output Table for Timing Pulses

Output: Execute

Inputs: Opcode / Current State	Load (000)	Store (001)	Add (010)	Sub (011)	Inc (100)	Dec (101)	Bra (110)	Beq (111)
000	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0
010	0	0	0	0	0	0	0	0
011	0	0	0	0	0	0	0	0
100	1	1	1	1	1	1	1	1
101	1	1	1	1	1	1	x	x
110	x	x	1	1	1	1	x	x
111	x	x	1	1	1	1	x	x

Figure 5: Output Table for Execute

A component for the FSM is shown in Figure 6 and verification results are shown in Figure 7.

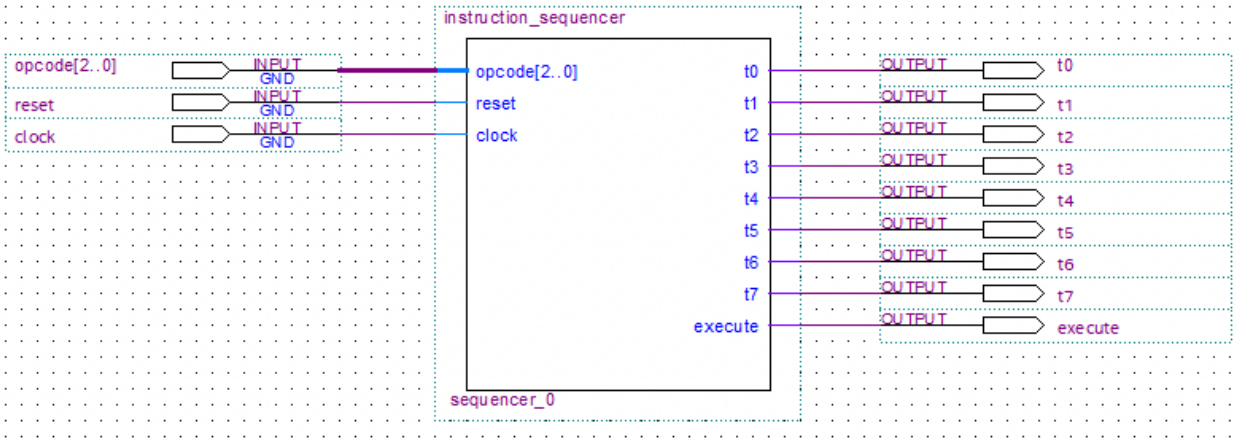

Figure 6: Instruction Sequencer FSM

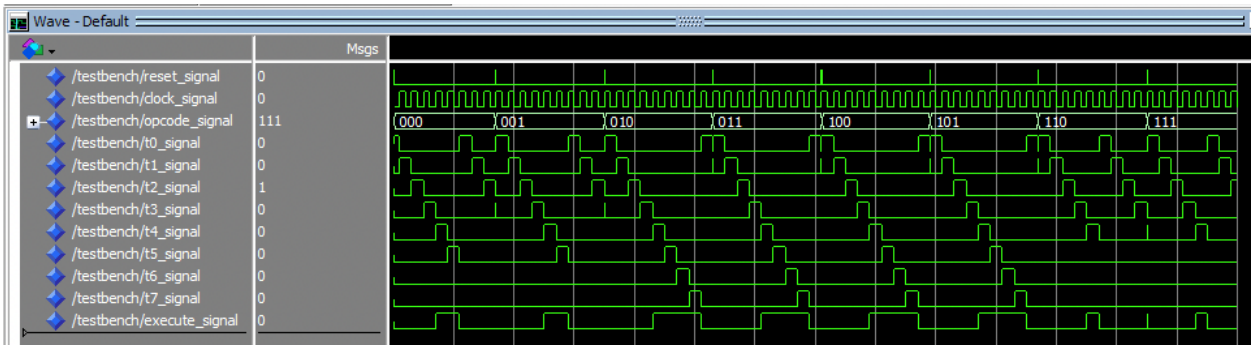

Figure 7: Test Results

What To Submit

You will submit the VHDL file you create as part of this assignment, along with simulation results for verification (as a PDF file, annotated appropriately where necessary).

Grading Rubric

This assignment is worth a total of 20 points:

- VHDL file of instruction sequencer design
- VHDL file of testbench
- Numerical Verification