Control Signals Logic

By Sidhardh Burre

CS/ECE 2330 Digital Logic Design

February 27, 2021

Table of Contents

1 Problem Statement 1
2 Analytical Design 2
3 Numerical Verification 5
4 Summary 6

Table of Figures

Figure 1: Simple CPU Architecture 1
Figure 2: Logical Expressions for Control Signals Logic 3
Figure 3: Circuits for Control Signals Logic 4
Figure 4: Circuit for Z storage 4
Figure 5: Input Test Vectors 5
Figure 6: Test Results (Table) 5
Figure 7: Test Results (Graph) 6

1 Problem Statement

Design circuits to generate the control signals for the Central Processing Unit (CPU) architecture shown in Figure 1. The signals are listed in Table 1, and the arithmetic operations defined by F0 and F1 are listed in Table 2.

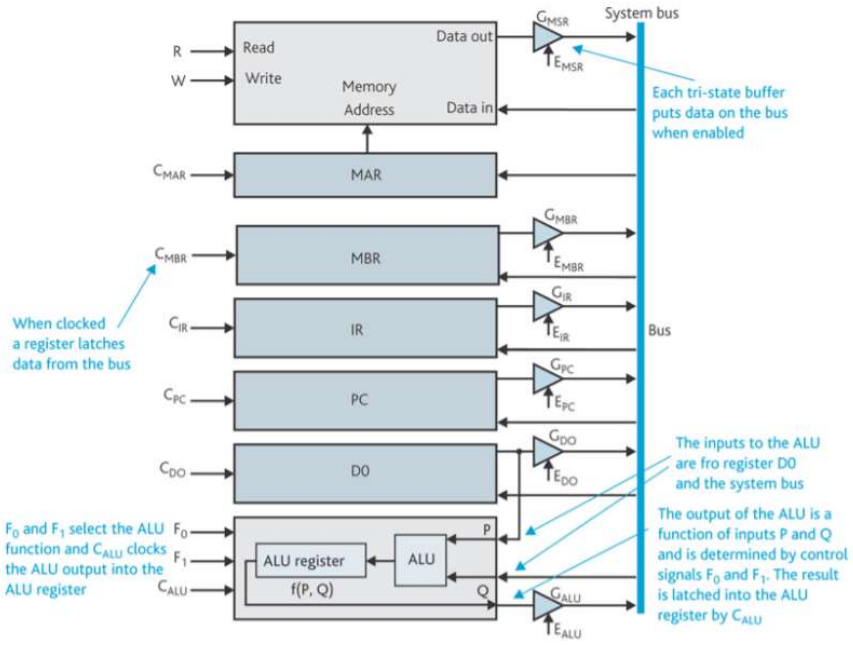

Figure 1: Simple CPU Architecture

Signal	Type	Operation
R	Memory Control	Read from memory
W	Memory Control	Write to memory
C_{MAR}	Register Clock	Clock data into MAR
C_{MBR}	Register Clock	Clock data into MBR
C_{PC}	Register Clock	Clock data into PC
C_{IR}	Register Clock	Clock data into IR
C_{D0}	Register Clock	Clock data into D0
C_{ALU}	Register Clock	Clock data into ALU register
E_{MSR}	Bus Control	Enable data from memory onto system bus
E_{MBR}	Bus Control	Enable data from MBR onto system bus
E_{PC}	Bus Control	Enable data from PC onto system bus
E_{IR}	Bus Control	Enable data from IR onto system bus
E_{D0}	Bus Control	Enable data from D0 onto system bus
E_{ALU}	Bus Control	Enable data from ALU register onto system bus
F_0	ALU Control	Select ALU function, bit 0
F_1	ALU Control	Select ALU function, bit 1

Table 1: Simple CPU Control Signals

F_1	F_0	Operation
0	0	Add P to Q, P + Q
0	1	Subtract Q from P, P – Q
1	0	Increment Q, Q + 1
1	1	Decrement Q, Q – 1

Table 2: Arithmetic Functions Defined by F0 and F1

The table in Figure 2 provides the Register Transfer Level (RTL) operations, or microinstructions, which indicate the sequence of operations that must occur for a given instruction is not executed. The sequencing of these microinstructions occurs within the timing signals, T0-T7. For example, the 4 microinstructions associated with the fetch operation are:

- MAR ← [PC]: The content of the PC is loaded into the MAR
- IR ← [MAR]: The memory content at the address contained in MAR is loaded into the IR
- ALU (Q) ←[PC]: The content of the PC is provided as the Q input to the ALU
- PC ← [ALU]: The output of the ALU is then loaded into the PC

		Instruction Sequence Timing Signals								Control Actions															
Instruction	Operations (RTL)	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	R E_{MSR}	W	C_{MAR}	C_{MBR}	C_{PC}	C_{IR}	C_{D0}	C_{ALU}	E_{MBR}	E_{PC}	E_{IR}	E_{D0}	E_{ALU}	F_1	F_0
Fetch	MAR ← [PC]	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
	IR ← [MAR]	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	ALU (Q) ← [PC]	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
	PC ← [ALU]	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
Load	MAR ← [IR]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	D0 ← [MAR]	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Store	MAR ← [IR]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	[MAR] ← [D0]	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Add	MAR ← [IR]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	MBR ← [MAR]	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	ALU (P) ← [MBR]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
	D0 ← [ALU]	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
Sub	MAR ← [IR]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	MBR ← [MAR]	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	ALU (P) ← [MBR]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
	D0 ← [ALU]	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
Inc	MAR ← [IR]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	MBR ← [MAR]	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	ALU (P) ← [MBR]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
	[MAR] ← [ALU]	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Dec	MAR ← [IR]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	MBR ← [MAR]	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	ALU (P) ← [MBR]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1
	[MAR] ← [ALU]	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Bra	PC ← [IR]	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
Beq	If Z = 1 then PC ← [IR]	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Figure 2: Instruction Sequencing and Control Signals

The Z signal is an output from the ALU and indicates whether the ALU operation produced a result equal to zero. Z=1 indicates that the ALU operation result equals zero; Z = 0 indicates otherwise. This signal is used by the **Beq** instruction to determine if the contents of the IR are loaded into the PC, thus executing what is known as a conditional branch. Because Z is used by a subsequent instruction, your control signals logic design should include a simple sequential logic circuit to store the value of Z after any ALU operation during the T6 instruction sequence period (that is, when T6 = 1).

2 Analytical Design

The logical equations for the Control Signals are shown in Figure 2 that allows for the Control Signal input to be converted into the corresponding command output.

All possible solutions

Read = (Add ∧ T5) ∨ (Dec ∧ T5) ∨ (Inc ∧ T5) ∨ (Load ∧ T5) ∨ (Sub ∧ T5) ∨ T1

Write = (Dec ∧ T7) ∨ (Inc ∧ T7) ∨ (Store ∧ T5)

Cmar = (Add ∧ T4) ∨ (Dec ∧ T4) ∨ (Inc ∧ T4) ∨ (Load ∧ T4) ∨ (Store ∧ T4) ∨ (Sub ∧ T4) ∨ T0

Cmbr = (Add ∧ T5) ∨ (Dec ∧ T5) ∨ (Inc ∧ T5) ∨ (Sub ∧ T5)

Embr = (Add ∧ T6) ∨ (Dec ∧ T6) ∨ (Inc ∧ T6) ∨ (Sub ∧ T6)

Cir = T1

Eir = (Add ∧ T4) ∨ (Beq ∧ T4) ∨ (Bra ∧ T4) ∨ (Dec ∧ T4) ∨ (Inc ∧ T4) ∨ (Load ∧ T4) ∨ (Store ∧ T4) ∨ (Sub ∧ T4)

Cpc = (Beq ∧ T4) ∨ (Bra ∧ T4) ∨ T3

Epc = T0 ∨ T2

Cd0 = (Add ∧ T7) ∨ (Load ∧ T5) ∨ (Sub ∧ T7)

Ed0 = Store ∧ T5

Calu = (Add ∧ T6) ∨ (Dec ∧ T6) ∨ (Inc ∧ T6) ∨ (Sub ∧ T6) ∨ T2

Ealu = (Add ∧ T7) ∨ (Dec ∧ T7) ∨ (Inc ∧ T7) ∨ (Sub ∧ T7) ∨ T3

F0 = (Dec ∧ T6) ∨ (Sub ∧ T6)

F1 = (Dec ∧ T6) ∨ (Inc ∧ T6) ∨ T2

Figure 3: Logical Expressions for Control Signals Logic

The logical expressions were analyzed to form the circuit schematic shown in Figure 3 and the Z value storage Figure 4.

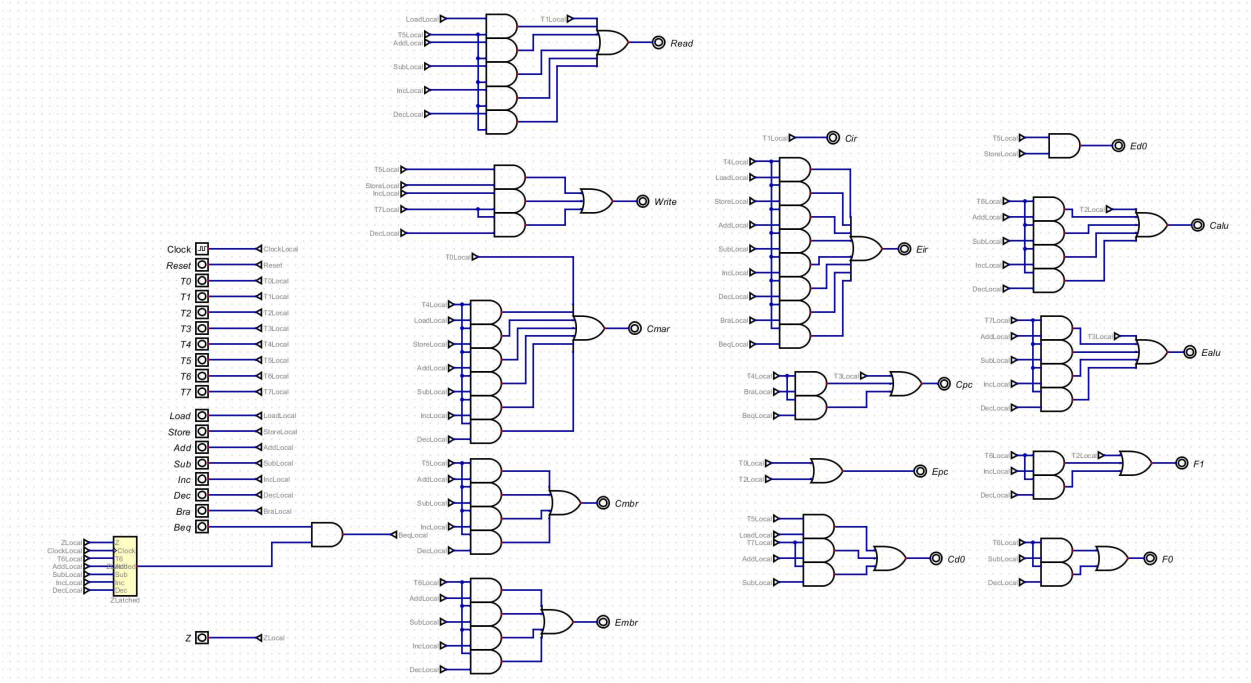

Figure 4: Circuits for Control Signals Logic

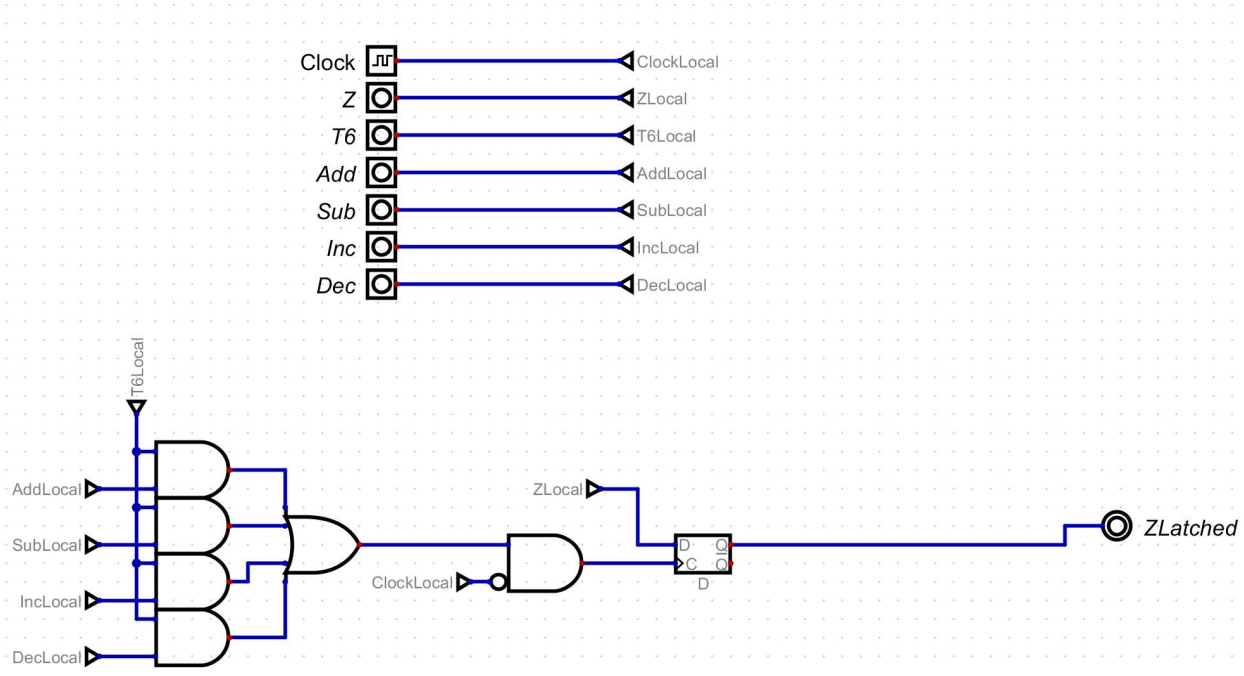

Figure 5: Circuit for Z storage

3 Numerical Verification

Shown in Figure 4, Figure 5, and Figure 6 are the numerical verification results, where an extensive set of input test vectors (Figure 4) were applied to the circuit.

Test data

```
# Test vectors for control signals logic
Reset Clock Load Store Add Sub Inc Dec Bra Beq T0 T1 T2 T3 T4 T5 T6 T7 Z Read Write Cmar Cmbr Cpc Cir Cd0 Calu Embr Epc Eir Ed0 Ealu F1 F0
# Load instruction
# Fetch: MAR ← [PC]
0     1     0    0     0   0   0   0   0   0   1  0  0  0  0  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0
0     0     0    0     0   0   0   0   0   0   1  0  0  0  0  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0
# IR ← [MAR]
0     1     0    0     0   0   0   0   0   0   0  1  0  0  0  0  0  0  0 1    0     0     0    0   1   0   0    0   0   0   0   0   0
0     0     0    0     0   0   0   0   0   0   0  1  0  0  0  0  0  0  0 1    0     0     0    0   1   0   0    0   0   0   0   0   0
# ALU (Q) ← [PC]
0     1     0    0     0   0   0   0   0   0   0  0  1  0  0  0  0  0  0 0    0     0     0    0   0   0   1    0   1   0   0   0   1   0
0     0     0    0     0   0   0   0   0   0   0  0  1  0  0  0  0  0  0 0    0     0     0    0   0   0   1    0   1   0   0   0   1   0
# PC ← [ALU]
0     1     0    0     0   0   0   0   0   0   0  0  0  1  0  0  0  0  0 0    0     0     0    1   0   0   0    0   0   0   0   1   0   0
0     0     0    0     0   0   0   0   0   0   0  0  0  1  0  0  0  0  0 0    0     0     0    1   0   0   0    0   0   0   0   1   0   0
# MAR ← [IR]
0     1     1    0     0   0   0   0   0   0   0  0  0  0  1  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0   0
0     0     1    0     0   0   0   0   0   0   0  0  0  0  1  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0   0
# D0 ← [MAR]
0     1     1    0     0   0   0   0   0   0   0  0  0  0  0  1  0  0  0 1    0     0     0    0   0   1   0    0   0   0   0   0   0   0
0     0     1    0     0   0   0   0   0   0   0  0  0  0  0  1  0  0  0 1    0     0     0    0   0   1   0    0   0   0   0   0   0   0
# Store instruction
# Rst Clock Load Store Add Sub Inc Dec Bra Beq T0 T1 T2 T3 T4 T5 T6 T7 Z Read Write Cmar Cmbr Cpc Cir Cd0 Calu Embr Epc Eir Ed0 Ealu F1 F0
# Fetch: MAR ← [PC]
0     1     0    0     0   0   0   0   0   0   1  0  0  0  0  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0
0     0     0    0     0   0   0   0   0   0   1  0  0  0  0  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0
# IR ← [MAR]
0     1     0    0     0   0   0   0   0   0   0  1  0  0  0  0  0  0  0 1    0     0     0    0   1   0   0    0   0   0   0   0   0
0     0     0    0     0   0   0   0   0   0   0  1  0  0  0  0  0  0  0 1    0     0     0    0   1   0   0    0   0   0   0   0   0
# ALU (Q) ← [PC]
0     1     0    0     0   0   0   0   0   0   0  0  1  0  0  0  0  0  0 0    0     0     0    0   0   0   1    0   1   0   0   0   1   0
0     0     0    0     0   0   0   0   0   0   0  0  1  0  0  0  0  0  0 0    0     0     0    0   0   0   1    0   1   0   0   0   1   0
# PC ← [ALU]
0     1     0    0     0   0   0   0   0   0   0  0  0  1  0  0  0  0  0 0    0     0     0    1   0   0   0    0   0   0   0   1   0   0
0     0     0    0     0   0   0   0   0   0   0  0  0  1  0  0  0  0  0 0    0     0     0    1   0   0   0    0   0   0   0   1   0   0
# MAR ← [IR]
0     1     0    1     0   0   0   0   0   0   0  0  0  0  1  0  0  0  0 0    0     1     0    0   0   0   0    0   1   0   0   0   0   0
```

Help Cancel OK

Figure 6: Input Test Vectors

passed

	Re...	Clo...	Load	Store	Add	Sub	Inc	Dec	Bra	Beq	T0	T1	T2	T3	T4	T5	T6	T7	Z	Read	Write	Cm...	Cmbr	Cpc	Cir	Cd0	Calu	Embr	Epc	Eir	Ed0	Ealu	F1	F0
L5	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L8	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
L9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
L11	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
L12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
L14	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
L15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
L17	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L20	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
L21	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
L25	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
L26	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
L28	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
L29	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
L31	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
L32	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
L34	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
L35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
L37	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L38	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L40	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
L41	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
L45	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
L46	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L48	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
L49	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
L51	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
L52	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
L54	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
L55	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
L57	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
L58	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Figure 7: Test Results (Table)

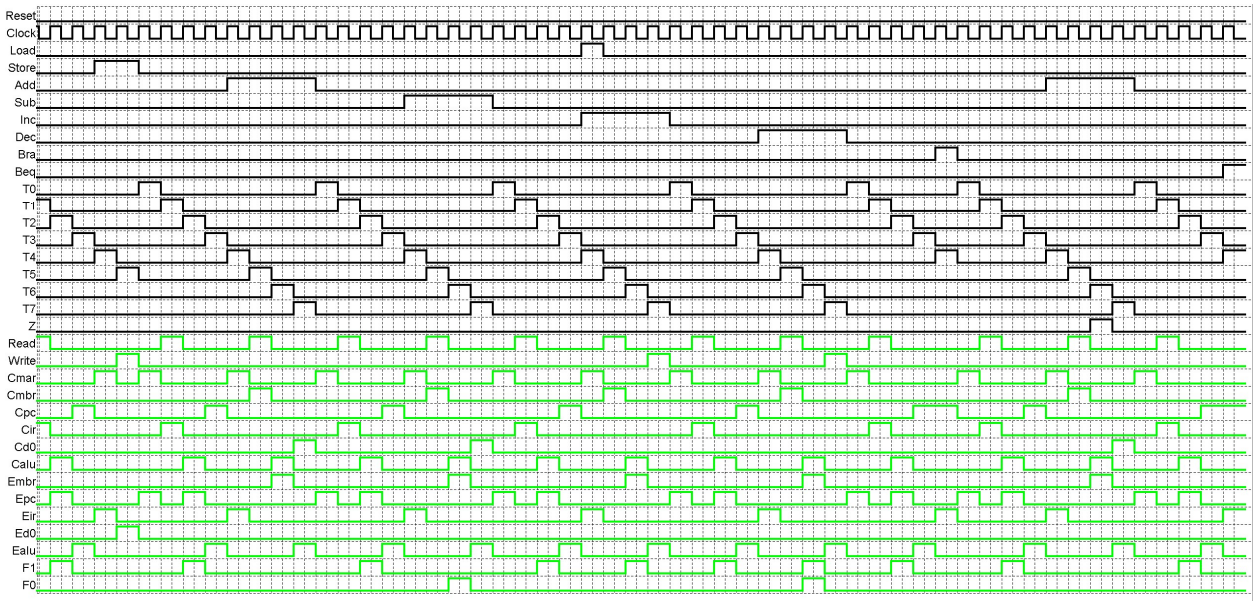

Figure 8: Test Results (Graph)

4 Summary

The problem requirement for this assignment was to design circuits to generate the control signals for the Central Processing Unit (CPU), inputs to the designed circuit are analyzed and output as a corresponding command to the CPU. Additionally, the Z input was to be stored during the T6 time period for determining the pursuant of the conditional or unconditional branch.

The truth table and corresponding logical equations were essential to defining functionality and developing a logic expression, that was then used to specify a digital circuit solution. The digital circuit was tested with an extensive set of input vectors (204 tests for 19 inputs) and produced the expected output for each test vector. Therefore, the specified digital circuit design solution satisfies the problem requirements.