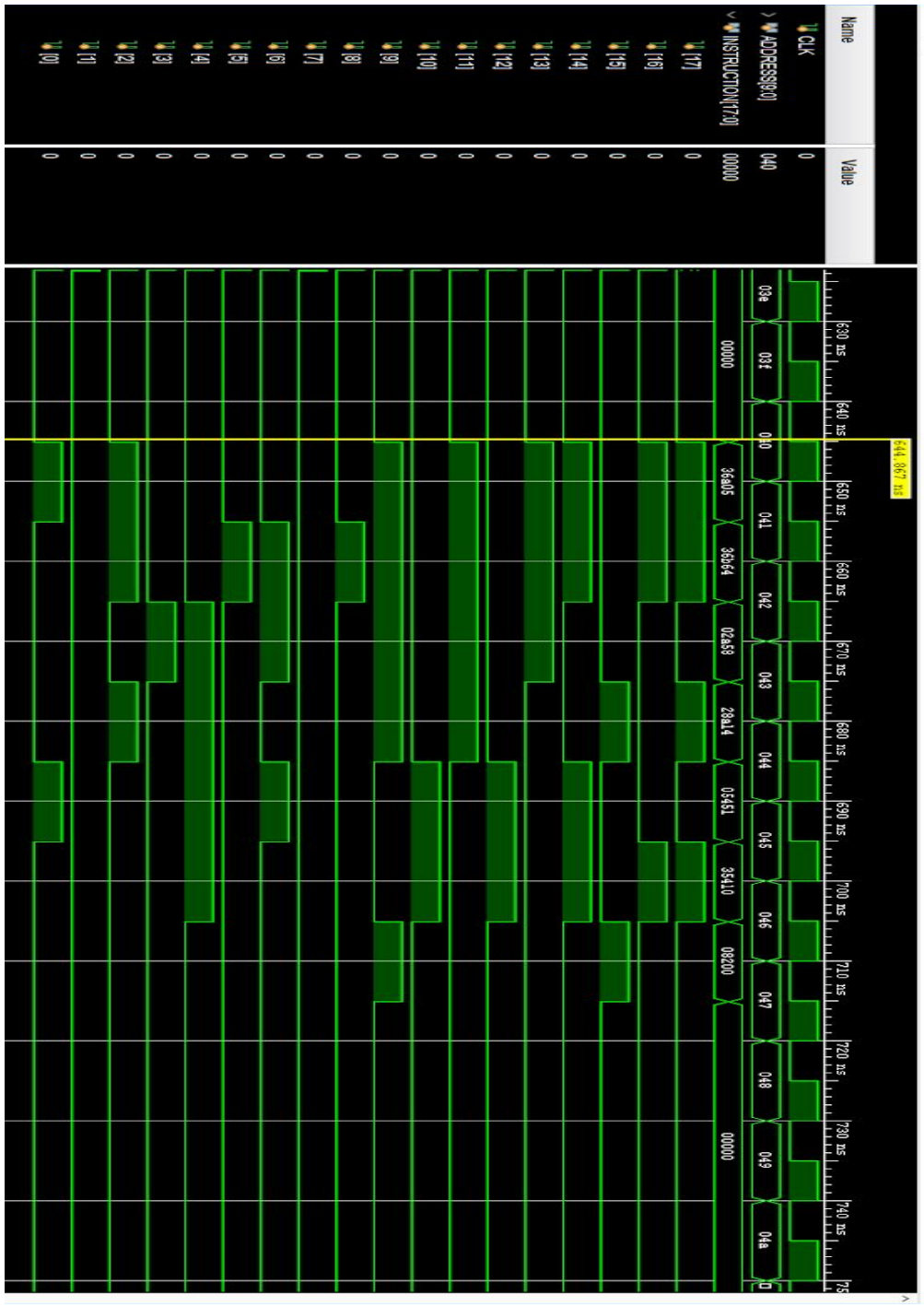
Part 1:

A. Program Analysis Table

address	instr	dest	c	z	out
0x40	mov R10, 0x05	R10 = 0x05	x	x	0x10 = 00
0x41	mov R11, 0x64	R11 = 0x64	x	x	0x10 = 00
0x42	add R10, R11	R10 = 0x69	0	0	0x10 = 00
0x43	add R10, 0x14	R10 = 0x7D	0	0	0x10 = 00
0x44	mov R20, R10	R20 = 0x7D	x	x	0x10 = 00
0x45	out R20, LED_PORT	x	x	x	0x10 = 7D

B. Simulation



Simulation of our first test bench.

In this test bench we wrote a for loop to loop through our test cases and create the chart above. The address is incrementing, and below it is the entire contents of the program starting at address 0x40.

Part 2:
A. Disassembly Table:

address	instruction	assembly	operands
0x40	0x37DFC	mov	R29, 0xFC
0x41	0x29D01	add	R29, 0x01
0x42	0x37EFA	mov	R30, 0xFA
0x43	0x37F05	mov	R31, 0x05
0x44	0x01EFA	exor	R30, R31
0x45	0x03EEA	sub	R30, R29
0x46	0x0820B	brne	0x41
0x47	0x00000	nop	

B. Typed Code:

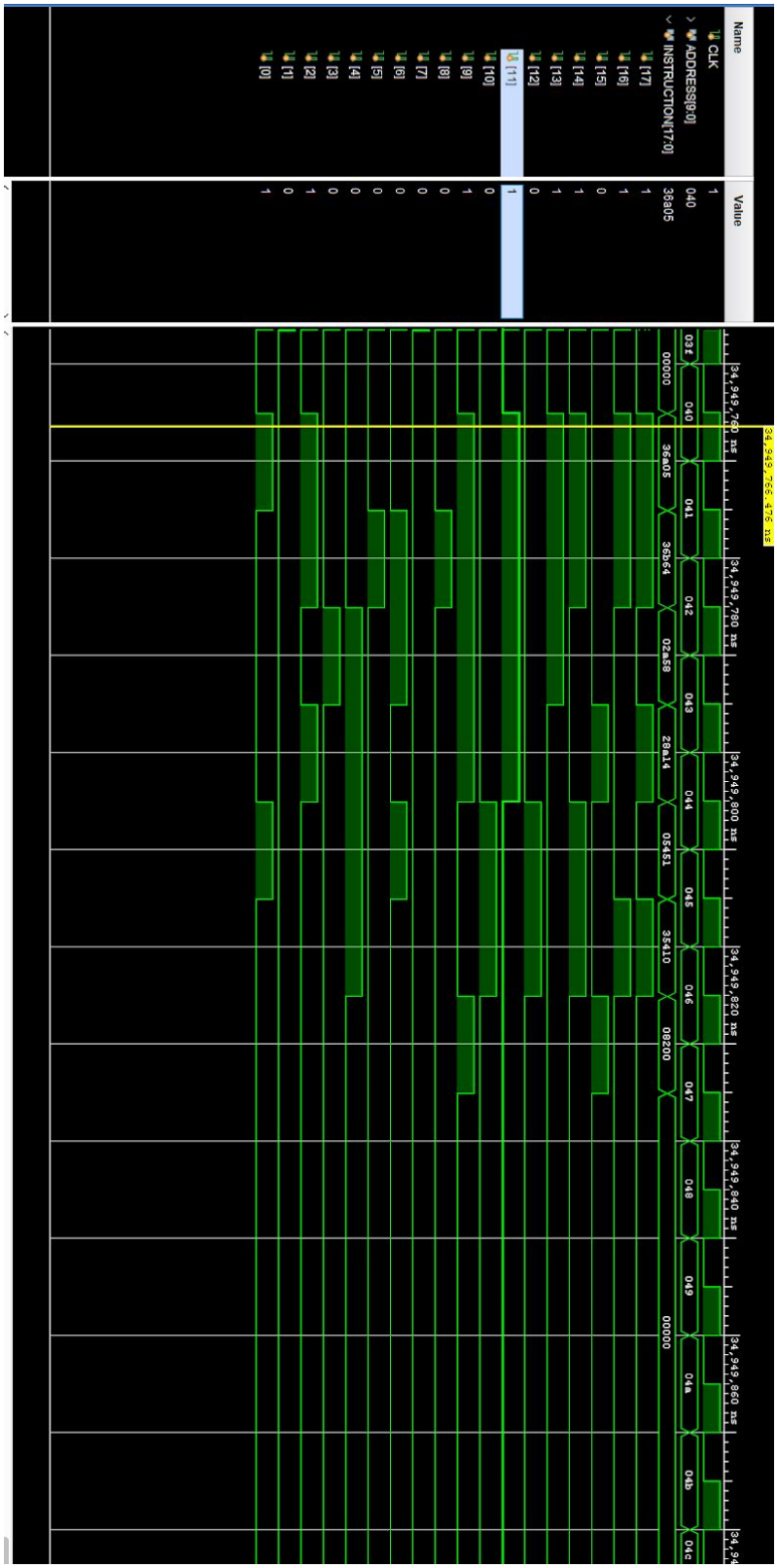
```
.EQU LED_PORT = 0x10      ; port for output
.CSEG
.ORG  0x40                ; code starts here

main_loop:          MOV   R29, 0xFC
                          ADD         R29, 0x01
                          MOV         R30, 0xFA
                          MOV         R31, 0x05
                          EXOR  R30, R31
                          SUB         R30, R29
                          BRNE  0x41
```

C. Table 3 for program A

address	instr	dest	c	z	out
0x40	mov R29, 0xFC	R29 = 0xFC	x	x	x
0x41	add R29, 0x01	R29 = 0xFD	0	0	x
0x42	Mov R30, 0xFA	R30 = 0xFA	x	x	x
0x43	Mov R31, 0x05	R31 = 0x05	x	x	x
0x44	Exor R30, R31	R30 = 0xFF	x	0	x
0x45	Sub R30, R29	R30 = 0x02	0	0	x
0x46	Brne 0x41	PC = 0x41	x	x	x
0x47	Nop	x	x	x	x

D. Simulation:

In this simulation we used a for loop to go through our test cases and create the chart below. The address is incrementing, and below it is the entire contents of the program starting at address 0x40.



Simulation of our second test bench.