HO CHI MINH UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

Assignment

Logic Design Project

Instructor:	Nguyen Tran Huu Nguyen
Group members:	Dao Hoanh Dat - 1752157
	Do Nguyen Dat - 1752160
	Nguyen Huu Anh Duc - 1752176
	Le Xuan Nguyen - 1752384

HO CHI MINH, 2020

Contents

1 Introduction to Central Processing Unit (CPU)

A processor (CPU) is the logic circuitry that responds to and processes the basic instructions that drive a computer. The CPU is seen as the main and most crucial integrated circuitry (IC) chip in a computer, as it is responsible for interpreting most of computers commands. CPUs will perform most basic arithmetic, logic and I/O operations, as well as allocate commands for other chips and components running in a computer.

2 Simple Processor

2.1 With instruction inputs from switches

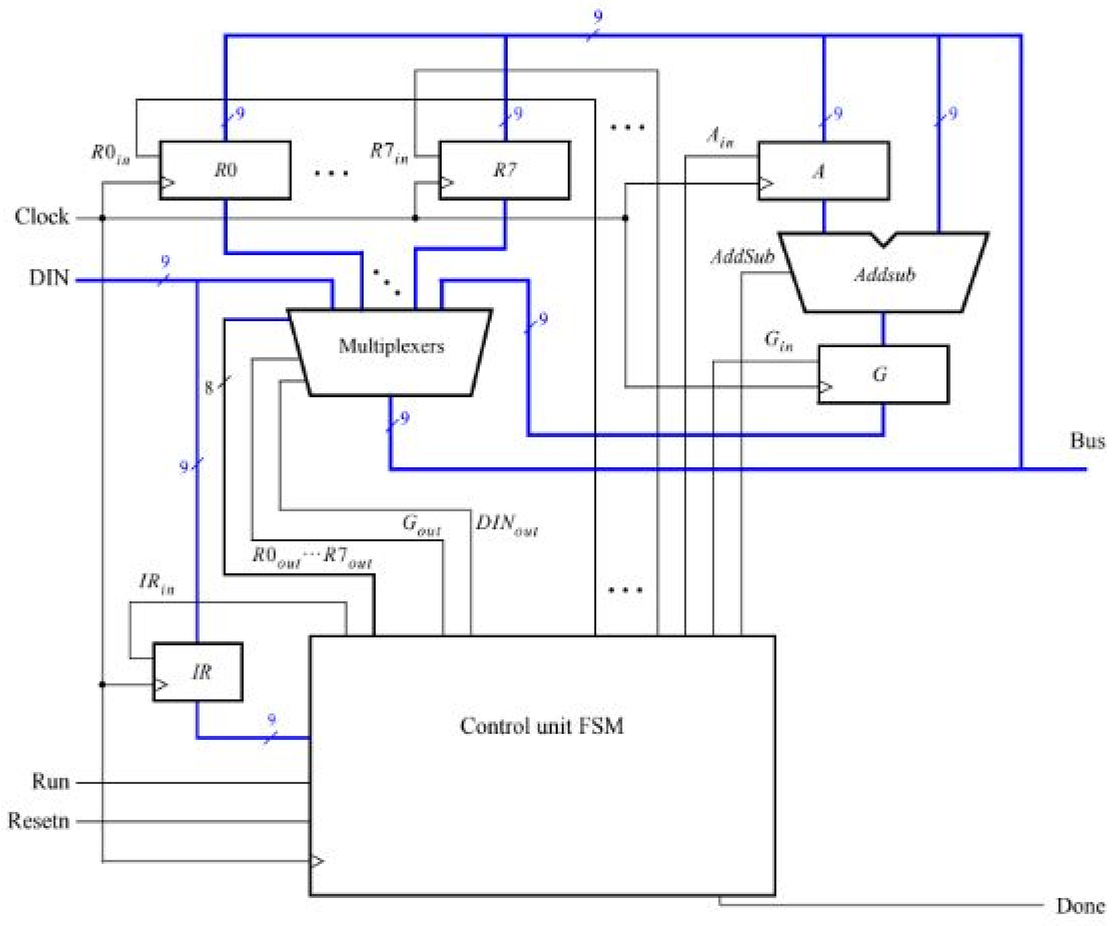
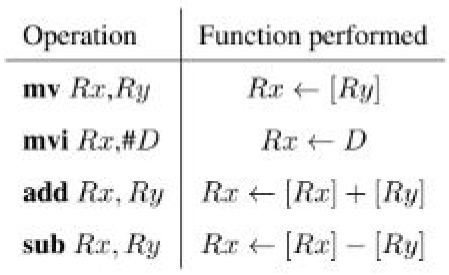


Figure 1: Diagram of Simple Processor



Operation	Function performed
mv Rx,Ry	$Rx \leftarrow [Ry]$
mvi $Rx,\#D$	$Rx \leftarrow D$
add Rx, Ry	$Rx \leftarrow [Rx] + [Ry]$
sub Rx, Ry	$Rx \leftarrow [Rx] - [Ry]$

Table 1: Instructions performed in the processor.

Each instruction can be encoded using the nine-bit format IIIXXXYYY, where III specifies the instruction, XXX gives the Rx register, and YYY gives the Ry register.III = 000 for the mv instruction, 001 for movi, 010 for add, and 011 for sub. Instructions are loaded from the the external input DIN, and stored into the IR register, using the connection indicated in Figure 1. For the mvi instruction the YYY field has no meaning, and the immediate data #D has to be supplied on the DIN input in the clock cycle after the mvi instruction word is stored into IR.

- Demo link: https://youtu.be/xWIOyvyQbe4
- Note:
  - SW[8:0]: input instruction into processor (DIN)
  - SW[9]: Run signal
  - KEY0: Reset signal
  - KEY1: Processor clock
  - KEY2: Choosing registers to show value
  - LEDR[8:0]: the instruction is being input
  - LEDG[8]: Done signal
  - HEX[2:0]: data of buswires
  - LEDR[17:15]: the register is being chosen to show value
  - HEX[6:4]: data of the chosen register
- Instructions are being used in the demo
  - 001000000 % move imediate to register 0 (mvi r0, #110(hex))
  - 100010000
  - 001001000 % move imediate to register 1 (mvi r1, #1b9(hex))
  - 110111001
  - 001010000 % move imediate to register 2 (mvi r2, #19(hex))
  - 000011001
  - 000011001 % move value(1b9(hex)) from register 1 to register 3 (mv r3, r1)
  - 010000010 % add register 0 and register 2 (add r0, r2)
  - 011001000 % subtract register 0 from register 1 (sub r1, r0)

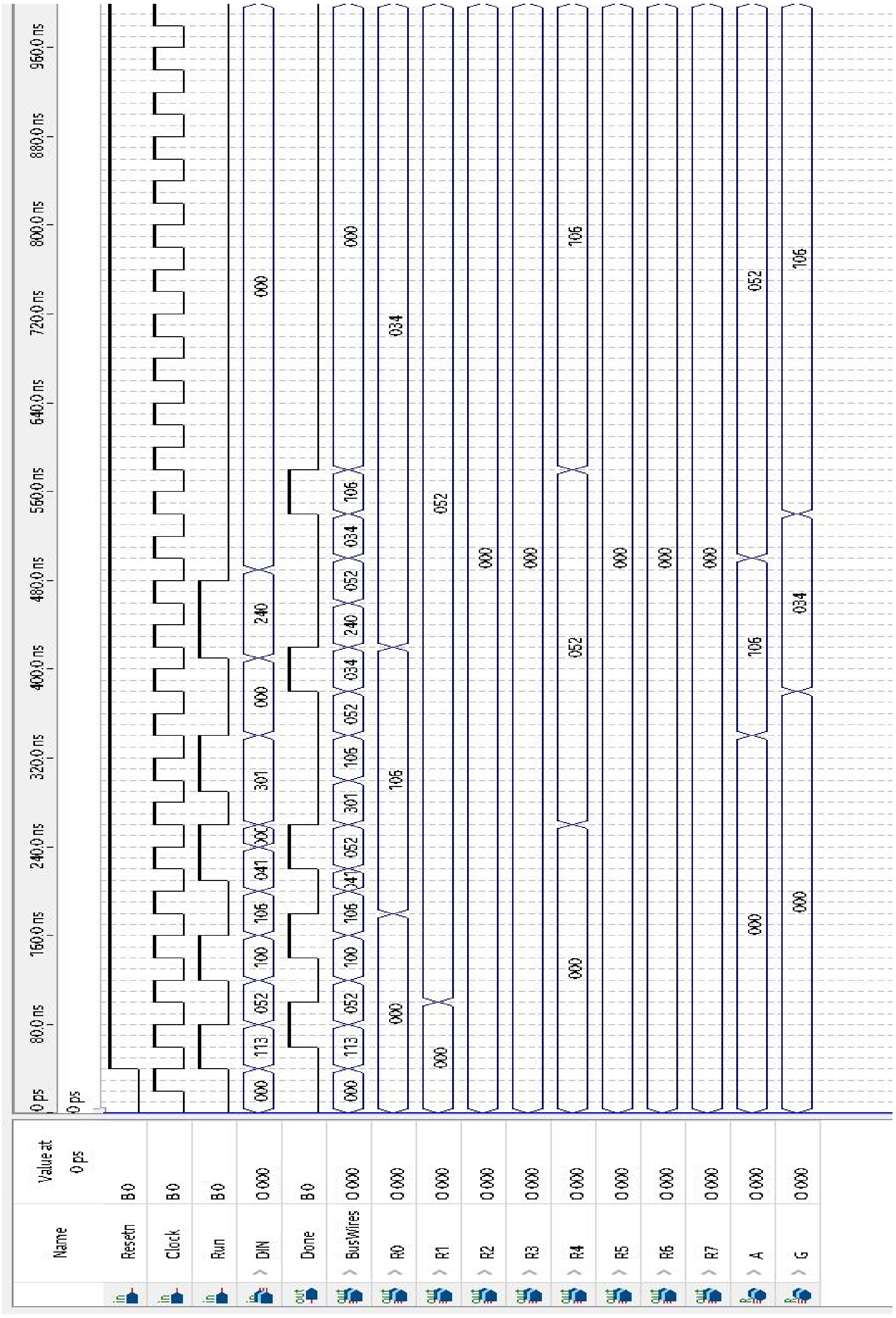


Figure 2: Waveform 2.1

2.2 With instruction inputs from memory

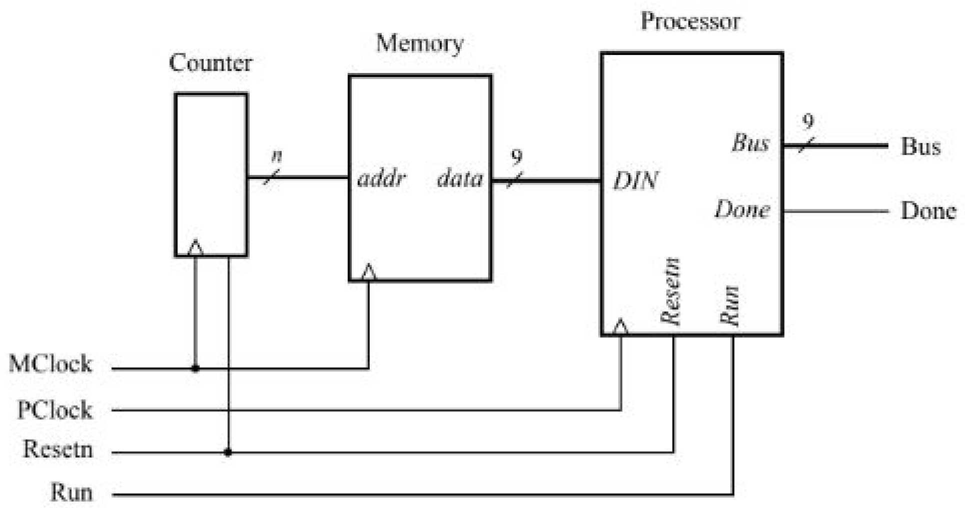


Figure 3: Diagram of Simple Processor

- Demo link: https://youtu.be/ZKZHmVwLdNI
- Note:
  - SW[9]: Run signal
  - KEY0: Reset signal
  - KEY1: Memory clock
  - KEY2: Processor clock
  - KEY3: Choosing registers to show value
  - LEDR[8:0]: the instruction is being input
  - LEDG[7:3]: Memory Address
  - LEDG[8]: Done signal
  - HEX[2:0]: data of buswires
  - LEDR[17:15]: the register is being chosen to show value
  - HEX[6:4]: data of the chosen register
- Instructions in the memory initialization file (mif)

```
WIDTH = 9;
DEPTH = 32;
ADDRESS_RADIX = UNS;
DATA_RADIX = BIN;

0 : 001000000;    % mvi r0, #5
```

```
1 : 000000101;
2 : 001001000;    % mvi r1, #81
3 : 010000001;
4 : 001010000;    % mvi r2, #F4
5 : 011110100;
6 : 000011001;    % mv r3, r1
7 : 010001000;    % add r1, r0
8 : 000100000;    % mv r4, r0
9 : 011010001;    % sub r2, r1
[10..31] : 000000000;
END;
```

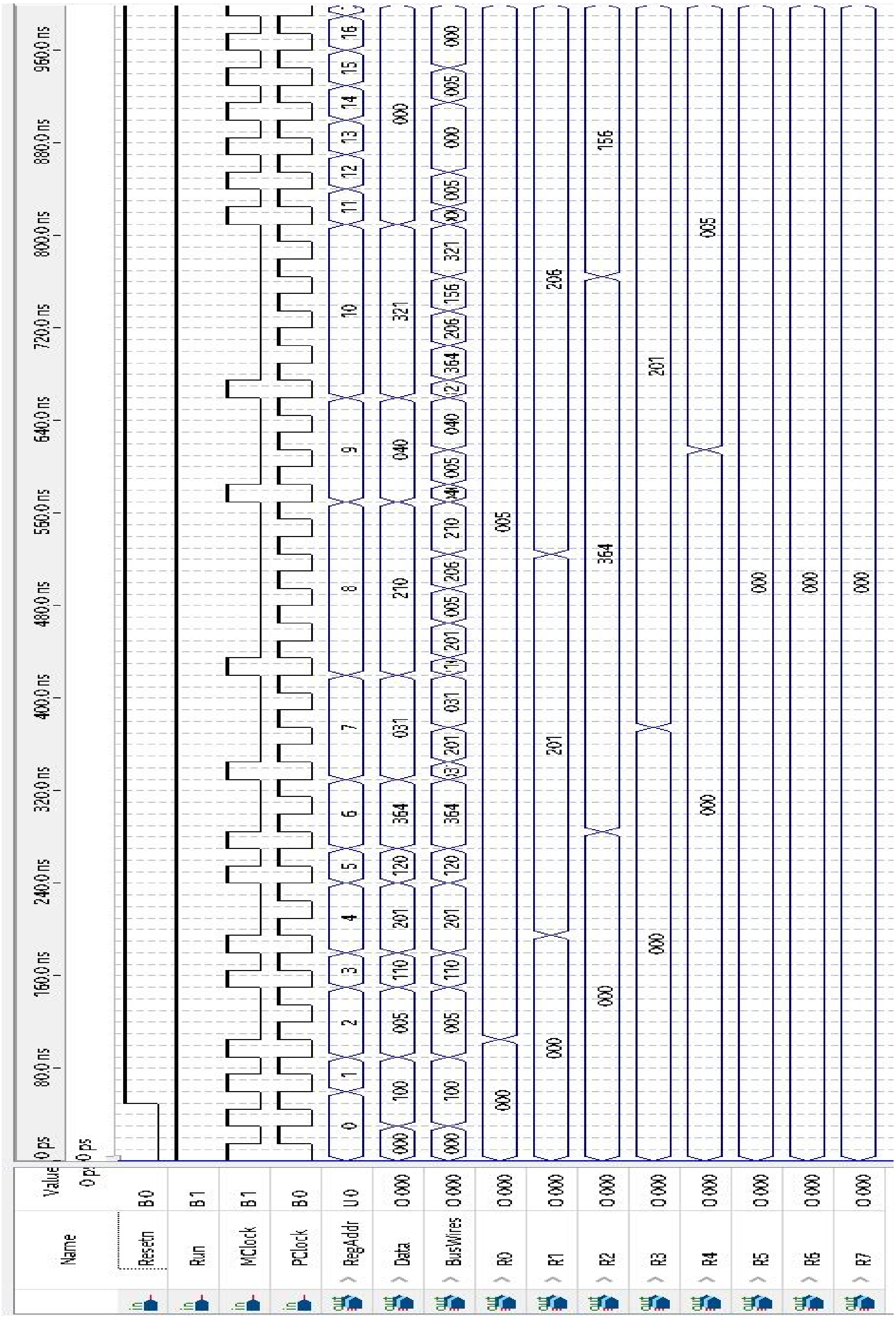


Figure 4: Waveform 2.2

References

[1] Stephen Brown and Zvonko Vranesic, *FUNDAMETALS OF DIGITAL LOGIC with Verilog Design*, The McGraw-Hill Companies, 2014.

[2] Intel Company, Digital Logic, `https://software.intel.com/content/www/us/en/develop/topics/fpga-academic/teach/digital-logic.html`