HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY – VNU HCMC
OFFICE FOR INTERNATIONAL STUDY PROGRAM
FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING

_____ * _____

DIGITAL SYSTEMS (LAB)
EXPERIMENTAL REPORT (PreLab 5)

Lecturer : **Mr. Nguyễn Tuấn Hùng**
Subject : **Digital Systems**
Class : **TT06**
Name : **Lương Triển Thắng**
Student ID : **2051194**

Ho Chi Minh City, 7th June, 2022

Contents

V Pre Laboratory 5

A simple processor

1. Known how to convert instructions to machine codes

a. Describe the instructions

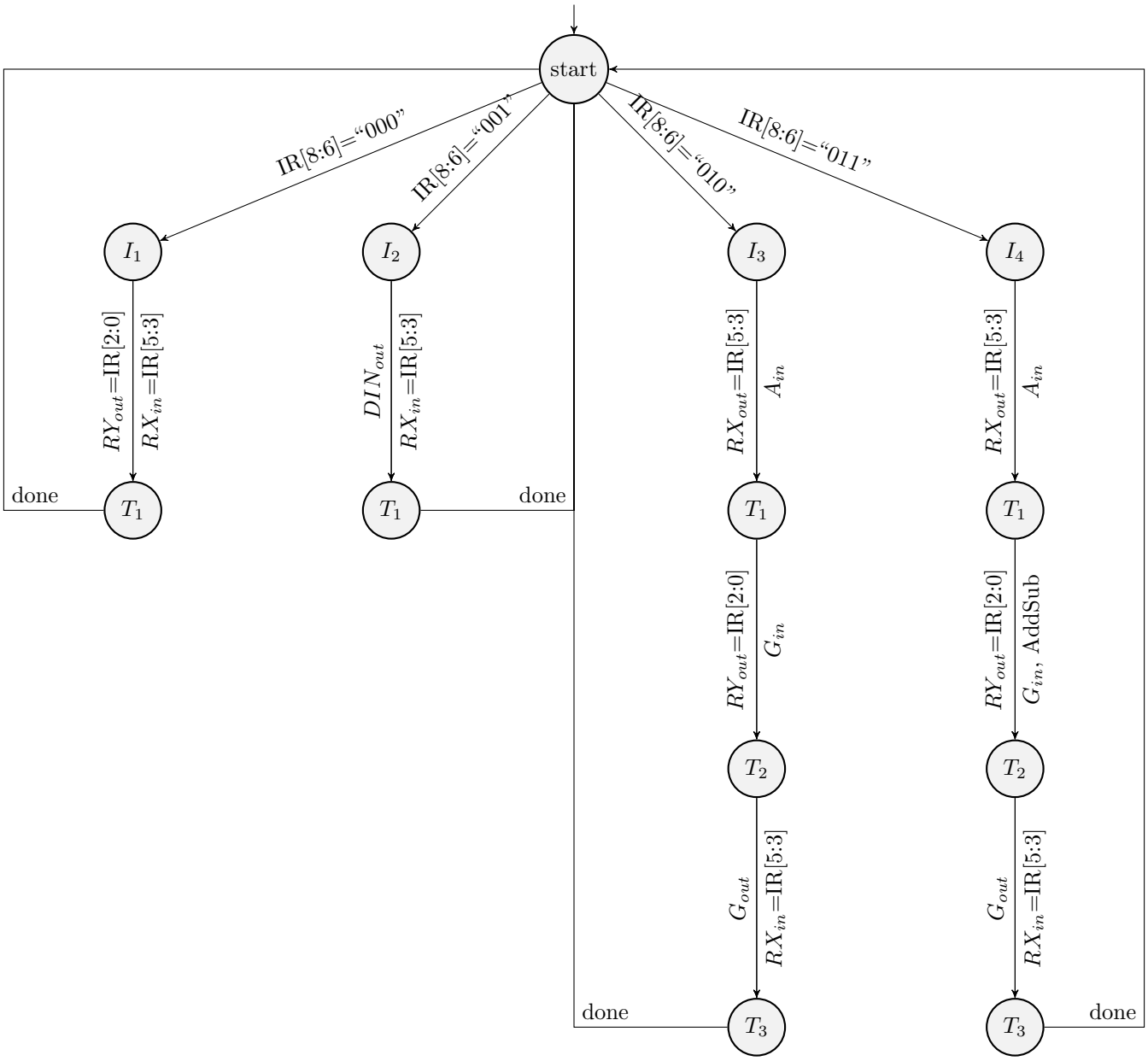
Instruction		Explain
mv	R1,R4	Move value in register R1 to R4
mv	R3,R2	Move value in register R3 to R2
mvi	R2,#5	Store 5 into register R2
mvi	R4,#-6	Store −6 into register R4
add	R3,R7	Add R3 and R7 and store into R3
add	R2,R0	Add R2 and R0 and store into R2
sub	R5,R6	Subtract R6 from R5 and store into R5

b. Intructions to machine codes

Instruction		Machine code			Notes
mv	R1,R4	000	001	100	
mv	R3,R2	000	011	010	
mvi	R2,#5	001	010	any	DIN = 000000101
mvi	R4,#-6	001	100	any	DIN = 111111010
add	R3,R7	010	011	111	
add	R2,R0	010	010	000	
sub	R5,R6	011	101	110	

2. Understand the control unit of the processor

FSM diagram and control signals of each state



3. Understand the memory block (ROM) organization

a. Code

myROM.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY myROM IS
	GENERIC (
		addr_width : INTEGER := 32; -- store 32 elements
		addr_bits : INTEGER := 5; -- required bits to store 32 elements
		data_width : INTEGER := 9 -- each element has 9-bits
	);
	PORT (
		addr : IN STD_LOGIC_VECTOR(addr_bits - 1 DOWNTO 0);
		data : OUT STD_LOGIC_VECTOR(data_width - 1 DOWNTO 0)
	);
END myROM;

ARCHITECTURE arch OF myROM IS

	TYPE rom_type IS ARRAY (0 TO addr_width - 1) OF STD_LOGIC_VECTOR(data_width - 1 DOWNTO 0);

	SIGNAL user_ROM : rom_type;

	-- note that 'ram_init_file' is not the user-defined name (it is attribute name)
	ATTRIBUTE ram_init_file : STRING;
	-- "rom_data.mif" is the relative address with respect to project directory
	-- suppose ".mif" file is saved in folder "ROM", then use "ROM/rom_data.mif"
	ATTRIBUTE ram_init_file OF user_ROM : SIGNAL IS "rom_data.mif";

BEGIN
	data <= user_ROM (to_integer(unsigned(addr)));
END arch;
```

Exc3.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;

ENTITY Exc3 IS
	PORT (
		clk : IN STD_LOGIC;
		SW : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
		LEDR : OUT STD_LOGIC_VECTOR(8 DOWNTO 0)
	);
END Exc3;

ARCHITECTURE arch OF Exc3 IS
	-- signal to store received data
	SIGNAL temp_data : STD_LOGIC_VECTOR (8 DOWNTO 0);
	SIGNAL temp_addr : STD_LOGIC_VECTOR(4 DOWNTO 0);
BEGIN
	user_ROM : ENTITY work.myROM PORT MAP (addr => temp_addr, data => temp_data);

	LEDR <= temp_data; -- display on LEDs

	PROCESS (clk)
	BEGIN
```

```
                IF rising_edge(clk) THEN
                        temp_addr <= SW;
                END IF;
        END PROCESS;
END arch;
```

rom_data.mif

```
WIDTH=9;
DEPTH=32;

ADDRESS_RADIX=UNS;
DATA_RADIX=BIN;

CONTENT BEGIN
        [0..15]  :   000000000;
        0 : 000001100;
        1 : 000011010;
        2 : 001010000;
        3 : 001100000;
        4 : 010011111;
        5 : 010010000;
        6 : 011101110;
END;
```

b. Waveform