ISTANBUL TECHNICAL UNIVERSITY COMPUTER ENGINEERING
DEPARTMENT

BLG222E
COMPUTER ORGANIZATION PROJECT 2
REPORT

Group Members:
150200011 Çağla Mıdıklı
150210049 Feyza Sarı
820210305 Berna Karatay

24 MAY 2023

Contents

1 Introduction **1**

2 Project or Materials and Methods **1**
2.1 Timer Module 1
2.2 Hardwired Control Unit 1
2.2.1 OPCODE 1
2.3 Cpu 2

3 Results **2**

4 Discussion **2**

5 Conclusion **2**

1 Introduction

In this project we designed hardwired control unit.

2 Project or Materials and Methods

2.1 Timer Module

The timer_4_bit module is a simple 4-bit timer that increments its output out on each positive clock edge. The value of out is controlled by the funsel input. When funsel is 0, out is set to 0, and when funsel is 1, out is incremented by 1.

2.2 Hardwired Control Unit

The hardwired control unit takes inputs from the ALU system and generates control signals and outputs for the computer. In this module inputs are reset, clock and IROut. ALU system inputs except RF_TSel are outputs to hardwired control unit: !!aluflag

- RF_OutASel, RF_OutBSel selects the source for Register File(RF) output A and B.
- RF_FunSel selects the function to perform on the RF output.
- RF_RSel selects the register to read from the RF.
- ALU_FunSel, ALUFlag selects the ALU function and determines if ALU flags should be updated.
- ARF_OutCSel, ARF_OutDSel selects the source for the Arithmetic Register File(ARF) output C and D.
- ARF_FunSel selects the function to perform on the ARF.
- ARF_RegSel selects the register to write to in the ARF.
- IR_LH, IR_Enable, IR_Funsel are control signals related to loading instructions from IR.
- Mem_CS, Mem_WR are control signals related to memory read/write operations.
- MuxASel, MuxBSel, MuxCSel are control signals for multiplexer selection.

In addition, there are output wires; addresing_mode represents the addressing mode of the instruction, regsel represents the register selection field of the instruction, op_address represents the operation address field of the instruction, operation_code represents the operation code field of the instruction. Destination_register, source_register1, source_register2 represents the register fields of the instruction. Timer_funsel and timer_out are control signals and output of the 4-bit timer.

2.2.1 OPCODE

First of all, we loaded the destination register by applying the logic operations specified in this part to the source registers selected with the code.

ALU Logical Operation Part(opcode 0,1,2,3,4,5,6): According to the 16-bit data input operation is chosen as the logic operation to be carried out in alu. According to the 16-bit data input, we pick the rf registers used as source registers. We use the operation inside the alu. Afterwards, we load it into the destination register (RF-ARF is an option).

Register Operation Part(opcode 7,8): Alu chooses to do the logic operation on itself. According to the 16-bit data input, we discover the rf registers used as source registers. We make no internal changes to the alu. Afterwards, we load it into the final register (RF-ARF is an option). and we change the register's funsel to either raise or decrease the register we loaded by 1.

Loading Operation Part(opcode 9,10,11,12,13): Without doing any logical processes, we load the value from our input or registers into the target register in this section. We choose our source register based on the input's format and the value between the fifth and eighth bits.

SP Operation Part(opcode 14,15):The final step in this process firstly involves loading our SP register into our destination register and then raising its value by 1 for opcode 14. And we add one to our sp register's value before loading it into our destination register for opcode 15.

2.3 Cpu

In this portion, we combined the hardware unit and the alu system we produced in this assignment to make a straightforward computer that is capable of carrying out specific tasks.

3 Results

By entering various values into a testbench that we built, we were able to determine whether our code functions properly. When the necessary memory entries are present, our CPU completes the operations we require correctly.

4 Discussion

Reset input is provided at simulation's beginning, clearing RF, ARF, and IR. The first instruction, BRA 0x20 0420, is then loaded from memory at the beginning. Because it is loaded in partial, less significant part loads first and then at the following clock more significant half part loads. and then the source registers and destination registers we choose are selected according to the inputs we receive from the memory.

5 Conclusion

The register structure, its kinds, the operation of the ALU system, and the hardware control unit of a computer are all covered in this assignment. The CPU's structure was learned by us. While developing the Alu System, we had the chance to observe logical expressions as well as other extreme conditions (overflow, barrow, carry).

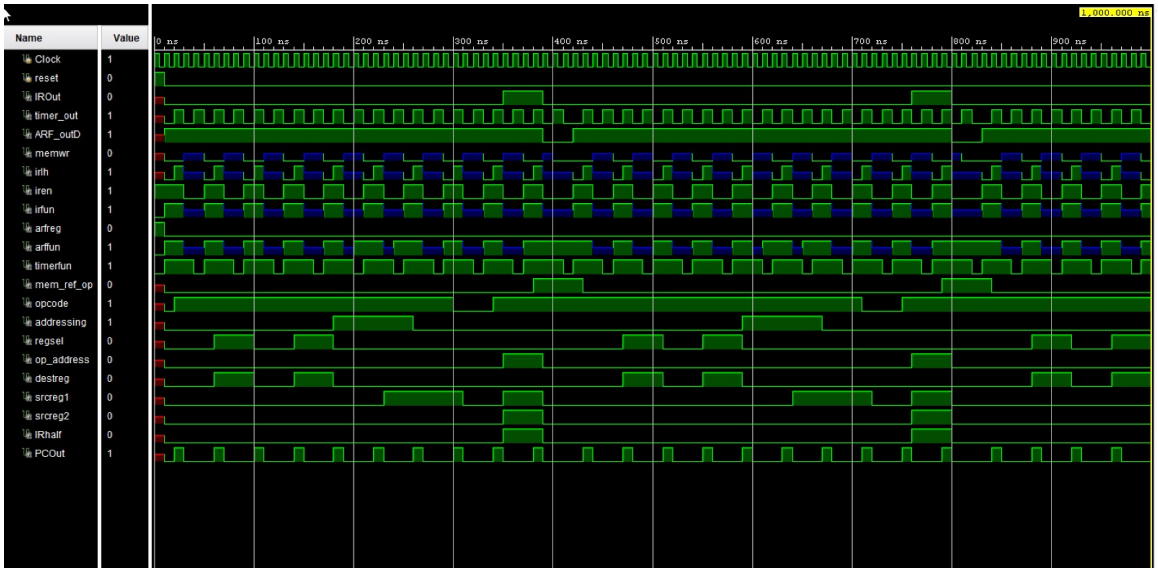

Figure 1: Figure for part-2a