ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 222E
COMPUTER ORGANIZATION
PROJECT 2 REPORT

CRN : 21334

LECTURER : Assoc. Prof. Gökhan İnce

GROUP MEMBERS:

150220916 : Rampıa Perente

150210020 : Abdullah Saatçi

SPRING 2024

Contents

1 INTRODUCTION [10 points]

During this project, we have experienced the design of a basic central processing unit (CPU). Using the Arithmetic Logic Unit System implemented in Project 1, different instructions that are given have been executed. Two types of instructions, one with address reference and the other without address reference have been designed according to descriptions and hex opcode. The instructions with the address reference have a 6-bit OPCODE that selects the operation, a 2-bit RSEL that selects the register, and an 8-bit ADDRESS. Other type of instructions have a 6-bit OPCODE that selects the operation, a 1-bit sign(S) that selects whether the flags will change or not, a 3-bit DESTREG that selects the destination register, a 3-bit SREG1 and a 3-bit SREG2 that select the source registers.

2 TASK DISTRIBUTION

The whole project has been done together.

3 MATERIALS AND METHODS [40 points]

3.1 MATERIALS

- Vivado

3.2 METHODS

3.2.1 INSTRUCTION TYPES

We have created a counter module to perform as SC in a regular digital circuit. With the help of this method, we were able to divide our operations into time cycles easily. The value of the counter will increase by 1 in each operation except the last one, where the reset of the counter will be activated, the counter will be reset to zero, and afterward, it will be ready to enter the cycle, again.

There are two types of instructions in this project, determined by distinct variables:
a) The first type of instruction has address reference and has the type of:

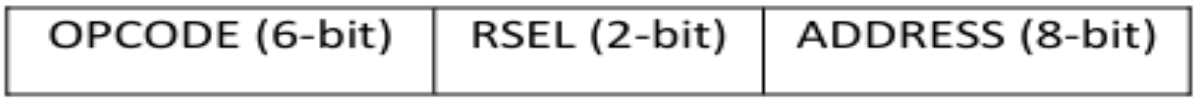

Figure 1: Instruction with Address Reference

Here, the RSEL selects the register that is going to be used, according to the table below:

RSEL	REGISTER
00	R1
01	R2
10	R3
11	R4

Table 1: Register Selection Table

```
case (RSEL)
    2'b00: RF_RegSel = 4'b0111;
    2'b01: RF_RegSel = 4'b1011;
    2'b10: RF_RegSel = 4'b1101;
    2'b11: RF_RegSel = 4'b1110;
endcase
```

Figure 2: Register Selection Example

b) The second type of instruction has address reference and has the type of:

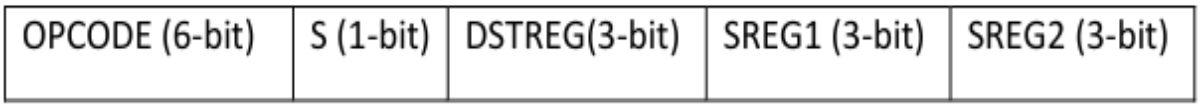

Figure 3: Instruction without Address Reference

Here, the DSTREG, SREG1, and SREG2 select the destination and the source registers that are going to be used, according to the table below:

DSTREG/SREG1/SREG2	REGISTER
000	PC
001	PC
010	SP
011	AR
100	R1
101	R2
110	R3
111	R4

Table 2: DSTREG, SREG1, and SREG2 Selection Table

```
case (DSTREG)
  3'b000: ARF_RegSel = 3'b011;
  3'b001: ARF_RegSel = 3'b011;
  3'b010: ARF_RegSel = 3'b110;
  3'b011: ARF_RegSel = 3'b101;
  3'b100: RF_RegSel = 4'b0111;
  3'b101: RF_RegSel = 4'b1011;
  3'b110: RF_RegSel = 4'b1101;
  3'b111: RF_RegSel = 4'b1110;
endcase
```

Figure 4: Destination Register Selection Example

Also, the 1-bit S selects whether the flags will change or not.

3.2.2 OPERATIONS

Before executing any operations, at the first clock cycle, we disabled all address registers(none of them is enabled) to not have any unexpected value and Program Counter(PC) is incremented by one, however, only the memory read is enabled when it is set to 0. Also, the instruction register's(IR) write is enabled to write the bits 7-0 of IR from memory. Then the counter is incremented to enter the next clock cycle.
At the second clock cycle, we selected PC with all selectors and increment it by one. Also, memory read and its chip select(CS) is enabled. Again, the instruction register's(IR) write is enabled to write the bits 15-8 of IR from memory. Then the counter is incremented to enter the next clock cycle.
From now on we can execute the instructions according to their OPCODE.

BRA

OPCODE: 000000
When the T is equal to 2, the CS of the memory and address registers are disabled, load of the R1 is enabled. Then the counter is incremented to enter the next clock cycle.
At the next clock cycle, load of R2 is enabled. Then the counter is incremented to enter the next clock cycle.
For the last clock cycle, no general purpose register is enabled, R1 and R2 are selected with OutASel and OutBSel, respectively, and they are holding PC and VALUE. Then addition operation enabled with ALU_Funsel to add PC and VALUE, afterward, the load is enabled to load them to PC and clock is reset.

BNE

OPCODE: 000001
At this operation when Z flag is equal to 0 the same operation as above is executed.

BEQ

OPCODE: 000010
At this operation when Z flag is equal to 1 the same operation as above is executed.

POP

OPCODE: 000011
When T is equal to 2, SP is selected with ARF_Regsel and Increment operation is selected with ARF_Funsel for SP + 1. Then, memory read and the register for R_x is selected for the assignment and load is enabled. Lastly, reset operation is done.

PSH

OPCODE: 000100

When T is equal to 2, the register for R_x is selected with RSEL and memory write, select and SP with OutDSel is enabled for assignment. When T is equal to 3, SP is selected with ARF_Regsel and Decrement operation is selected with ARF_Funsel for SP - 1. Lastly, memory read and select is enabled and reset operation is done.

INC

OPCODE: 000101

When T is equal to 2, the source register is selected with SREG1 and memory select is enabled. If the selected register is from Register File(RF) the assignment operetion to A done with ALU_Funsel. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and load is enabled. When T is equal to 3, the Increment operation is done according to the place of DESTREG. Lastly, reset operation is done.

DEC

OPCODE: 000110

When T is equal to 2, the source register is selected with SREG1 and memory select is enabled. If the selected register is from Register File(RF) the assignment operation to A done with ALU_Funsel. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and load is enabled. When T is equal to 4, the Decrement operation is done according to the place of DESTREG. Lastly, reset operation is done.

LSL

OPCODE: 000111

When T is equal to 2, the source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF) the logic shift left(LSL) operation done with ALU_Funsel. If the selected register is from Register File(RF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled. The LSL operation will be executed at the next time cycle. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 LSL operation is done. When T is equal to 4, according to the place of destination register all general purpose registers are disabled. Lastly, reset operation is done.

LSR

OPCODE: 001000

When T is equal to 2, the source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF) the logic shift right(LSR) operation done with ALU_Funsel. If the selected register is from Register File(RF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled. The LSR operation will be executed at the next time cycle. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 LSR operation is done. When T is equal to 4, according to the place of destination register all general purpose registers are disabled. Lastly, reset operation is done.

ASR

OPCODE: 001001

When T is equal to 2, all address registers are disabled, the source register is selected with SREG1, and memory select is disabled. If the selected register is from Address Register File(ARF) the arithmetic shift right(ASR) operation done with ALU_Funsel. If the selected register is from Register File(RF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled. The ASR operation will be executed at the next time cycle. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 ASR operation is done. When T is equal to 4, according to the place of destination register all general purpose registers are disabled. Lastly, reset operation is done.

CSL

OPCODE: 001010

When T is equal to 2, the source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF) the circular shift right(CSL) operation done with ALU_Funsel. If the selected register is from Register File(RF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled. The CSL operation will be executed at the next time cycle. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 CSL operation is done. When T is equal to 4, according to the place of destination register all general purpose registers are disabled. Lastly, reset operation is done.

CSR

OPCODE: 001011

When T is equal to 2, the source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF) the circular shift right(CSR) operation done with ALU_Funsel. If the selected register is from Register File(RF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled. The CSR operation will be executed at the next time cycle. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 CSR operation is done. When T is equal to 4, according to the place of destination register all general purpose registers are disabled. Lastly, reset operation is done.

AND

OPCODE: 001100

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 AND operation is done. Lastly, reset operation is done.

ORR

OPCODE: 001101

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 AND operation is done. Lastly, reset operation is done.

NOT

OPCODE: 001110

When T is equal to 2, the source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF) the complement(NOT) operation done with ALU_Funsel. If the selected register is from Register File(RF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled. The CSR operation will be executed at the next time cycle. When T is equal to 3, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 NOT operation is done. When T is equal to 4, according to the place of destination register all general purpose registers are disabled. Lastly, reset operation is done.

XOR

OPCODE: 001111

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 XOR operation is done. Lastly, reset operation is done.

NAND

OPCODE: 010000

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 NAND operation is done. Lastly, reset operation is done.

LDR(16-bit)

OPCODE: 010010

When T is equal to 2, AR is selected with ARF_OutDSel, MemoryOutput is selected with MuxASel, the source register is selected with RSEL and then load operation is done. Lastly, reset operation is done.

STR(16-bit)

OPCODE: 010011

When T is equal to 2, the source register is selected with RSEL and assignment operation is done. Then memory write and select is enabled and AR is selected with ARF_OutDSel to write to memory. Lastly, reset operation is done.

ADD

OPCODE: 010101

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 Addition operation is done. Lastly, reset operation is done.

ADS

OPCODE: 010110

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 Addition with carry operation is done. Lastly, reset operation is done.

SUB

OPCODE: 010111

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 Substraction operation is done. Lastly, reset operation is done.

MOVS

OPCODE: 011000

When T is equal to 2, the source register is selected with SREG1, memory select is disabled, and if sign bit(S) is 1 ALU_WF is enabled for flags. If the selected register is from Register File(RF) assignment operation is selected. When T is equal to 3, if sign bit(S) is 1 ALU_WF is enabled for flags and then, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 assignment operation is done. Lastly, reset operation is done.

ADDS

OPCODE: 011001

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4,if sign bit(S) is 1 ALU_WF is enabled and then, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 addition operation is done. Lastly, reset operation is done.

SUBS

OPCODE: 011010

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4,if sign bit(S) is 1 ALU_WF is enabled and then, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 substraction operation is done. Lastly, reset operation is done.

ANDS

OPCODE: 011011

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4,if sign bit(S) is 1 ALU_WF is enabled and then, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 AND operation is done. Lastly, reset operation is done.

ORRS

OPCODE: 011100

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4,if sign bit(S) is 1 ALU_WF is enabled and then, the destination register

is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 ORR operation is done. Lastly, reset operation is done.

XORS

OPCODE: 011101

When T is equal to 2, first source register is selected with SREG1 and memory select is disabled. If the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 3, second source register is selected with SREG2 and if the selected register is from Address Register File(ARF), firstly, it passes from MuxASel than we select a register for storage and than load is enabled than the selected register will be selected for output with OutBSel. When T is equal to 4,if sign bit(S) is 1 ALU_WF is enabled and then, the destination register is selected with DESTREG and the place of DESTREG and SREG1 is checked whether they are in ARF or RF and according to place of SREG1 XOR operation is done. Lastly, reset operation is done.

BX

OPCODE: 011110

When T is equal to 2, PC is enabled with ARF_OutCSel, memory read and select is enabled, and MemoryOut is selected with MuxASel. Then, S1 is enabled and loaded and subsequently it is selected as output with RF_OutASel and assignment operation is selected with ALU_FunSel. When T is equal to 3, SP is enabled with ARF_OutDSel and memory write and select is enabled. When T is equal to 4, SP is enabled with ARF_OutDSel and memory read is enabled, and select is disabled. Then, R_x is selected with RSEL, ALUOut is enabled with MuxBSel, and PC is enabled, then load operation is done and clock is reset.

BL

OPCODE: 011111

When T is equal to 2, SP is enabled with ARF_OutDSel, memory read and select is enabled, and MemoryOut is selected. When T is equal to 3, PC and load are enabled and memory select is disabled load operation is done and the clock is reset.

LDRIM

OPCODE: 100000

When T is equal to 2, the bits 7-0 of IR are selected with MuxASel, a register is selected with RSEL and memory select is disabled. Then, the load operation is done and the clock

is reset.

4 RESULTS [15 points]

This is our simulation result:

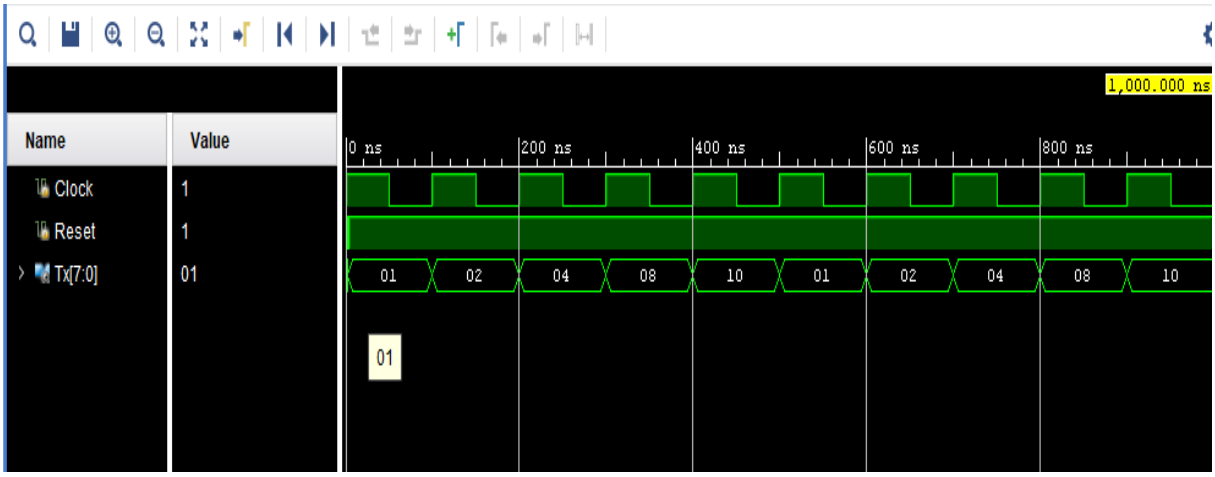

Figure 5: Simulation Output

This is one of the sample outputs:

```
Output Values:
T:   1
Address Register File: PC:     0, AR:     0, SP:     0
Instruction Register :     x
Register File Registers: R1:     0, R2:     0, R3:     0, R4:     0
Register File Scratch Registers: S1:     0, S2:     0, S3:     0, S4:     0
ALU Flags: Z: x, N: x, C: x, O: x
ALU Result: ALUOut:     x


Output Values:
T:   2
Address Register File: PC:     1, AR:     0, SP:     0
Instruction Register :     X
Register File Registers: R1:     0, R2:     0, R3:     0, R4:     0
Register File Scratch Registers: S1:     0, S2:     0, S3:     0, S4:     0
ALU Flags: Z: x, N: x, C: x, O: x
ALU Result: ALUOut:     x


Output Values:
T:   4
Address Register File: PC:     2, AR:     0, SP:     0
Instruction Register :     0
Register File Registers: R1:     0, R2:     0, R3:     0, R4:     0
Register File Scratch Registers: S1:     0, S2:     0, S3:     0, S4:     0
ALU Flags: Z: x, N: x, C: x, O: x
ALU Result: ALUOut:     x
```

Figure 6: Sample Output

5 DISCUSSION [25 points]

The project aimed at a basic hardwired control unit, which consists of different components coming from the Arithmetic Logic Unit System. The files from the previous project were used, also, CPUSystem.v was implemented. The instructions that have a hex OPCODE, an RSEL, and an address value are the BRA, BNE, BEQ, POP, PSH, MOVH, MOVL, LDR, STR, BX, BL, LDRIM, and STRIM instructions. The instructions that have a hex OPCODE, a sign bit, a DESTREG, a SREG1, and a SREG2 are the INC, DEC, LSL, LSR, ASR, CSL, CSR, AND, ORR, XOR, NOT, NAND, ADD, ADC, SUB, MOVS, ADDS, SUBS, ANDS, ORRS, and XORS instructions. What we have done for each instruction is explained in the METHODS section and the result of simulation is described in the RESULTS section. Overall, the project provided an invaluable opportunity to learn computer organization and digital design, empowering the team with skills crucial in this domain.

6 CONCLUSION [10 points]

Throughout the project, we gained a lot of knowledge about the use of the CPU and the implementation of a control unit that executes different types of instructions according to OPCODE. However, we had trouble determining within how many clock signals the given operations were going to be executed. Also, selecting registers according to the RSEL, the DSTREG, the SREG1, and the SREG2 was quite challenging due to the long lines of code.

[1]

REFERENCES

[1] Darte et al. Vivado. https://www.xilinx.com/developer/authors.html.