Department of Computer Engineering

BLG 222E
Computer Organization

Project Report

Project *:* 3

Date *:* 17.05.2017

Group Members:

ID	Name	Surname
150130029	*Muhammed Emin*	*Küçük*
040140003	*Burak Mert*	*Gönültaş*
150140025	*Elif*	*Kılıç*
150150701	*Yunus*	*Güngör*

1. INTRODUCTION

In this project, we designed a hardware control unit according to the required architecture; the control unit was built as an extension for the circuit from the second assignment.

2. REQUIREMENTS

INSTRUCTION FORMAT

Instructions with address reference

- The OPCODE is a 5-bit field (See Table 1 for the definition).
- The REGSEL is a 2-bit field (See left side of Table 2 for the definition).
- The ADDRESSING MODE is a 1-bit field (See Table 3 for the definition).
- The ADDRESS is 8 bits

Figure 1: Instructions with an address reference

Instructions without address reference

- The OPCODE is a 5-bit field (See Table 1 for the definition).
- DESTREG is a 3-bit field which specifies the destination register (See right side of Table 2 for the definition).
- SRCREG1 is a 3-bit field which specifies the first source register (See right side of Table 2 for the definition).
- SRCREG2 is a 3-bit field which specifies the second source register (See right side of Table 2 for the definition).

Figure 2: Instructions without an address reference

Table 1: OPCODE field and SYMBols for opertions and their descriptions

OPCODE (HEX)	SYMB	ADDRESSING MODE	DESCRIPTION
0x00	LD	IM, D	Rx ← Value (Value is described in Table 3)
0x01	ST	D	Value ← Rx
0x02	MOV	N/A	DESTREG ← SRCREG1
0x03	PSH	N/A	M[SP] ← Rx, SP ← SP - 1
0x04	PUL	N/A	SP ← SP + 1, Rx ← M[SP]
0x05	ADD	N/A	DESTREG ← SRCREG1 + SRCREG2
0x06	SUB	N/A	DESTREG ← SRCREG2 - SRCREG1
0x07	DEC	N/A	DESTREG ← SRCREG1 - 1
0x08	INC	N/A	DESTREG ← SRCREG1 + 1
0x09	AND	N/A	DESTREG ← SRCREG1 AND SRCREG2
0x0A	OR	N/A	DESTREG ← SRCREG1 OR SRCREG2
0x0B	NOT	N/A	DESTREG ← NOT SRCREG1
0x0C	LSL	N/A	DESTREG ← LSL SRCREG1
0xOD	LSR	N/A	DESTREG ← LSR SRCREG1
0x0E	BRA	IM	PC ← Value
0x0F	BEQ	IM	IF Z=1 THEN PC ← Value
0x10	BNE	IM	IF Z=0 THEN PC ← Value
0x11	CALL	IM	M[SP] ← PC, SP ← SP – 1, PC ← Value
0x12	RET	N/A	SP ← SP + 1, PC ← M[SP]

Table 2:REGSEL (Left) and DESTREG/SRCREG1/SRCREG2 (Right) select the register of interest for a particular instruction

REGSEL	REGISTER
00	R0
01	R1
10	R2
11	R3

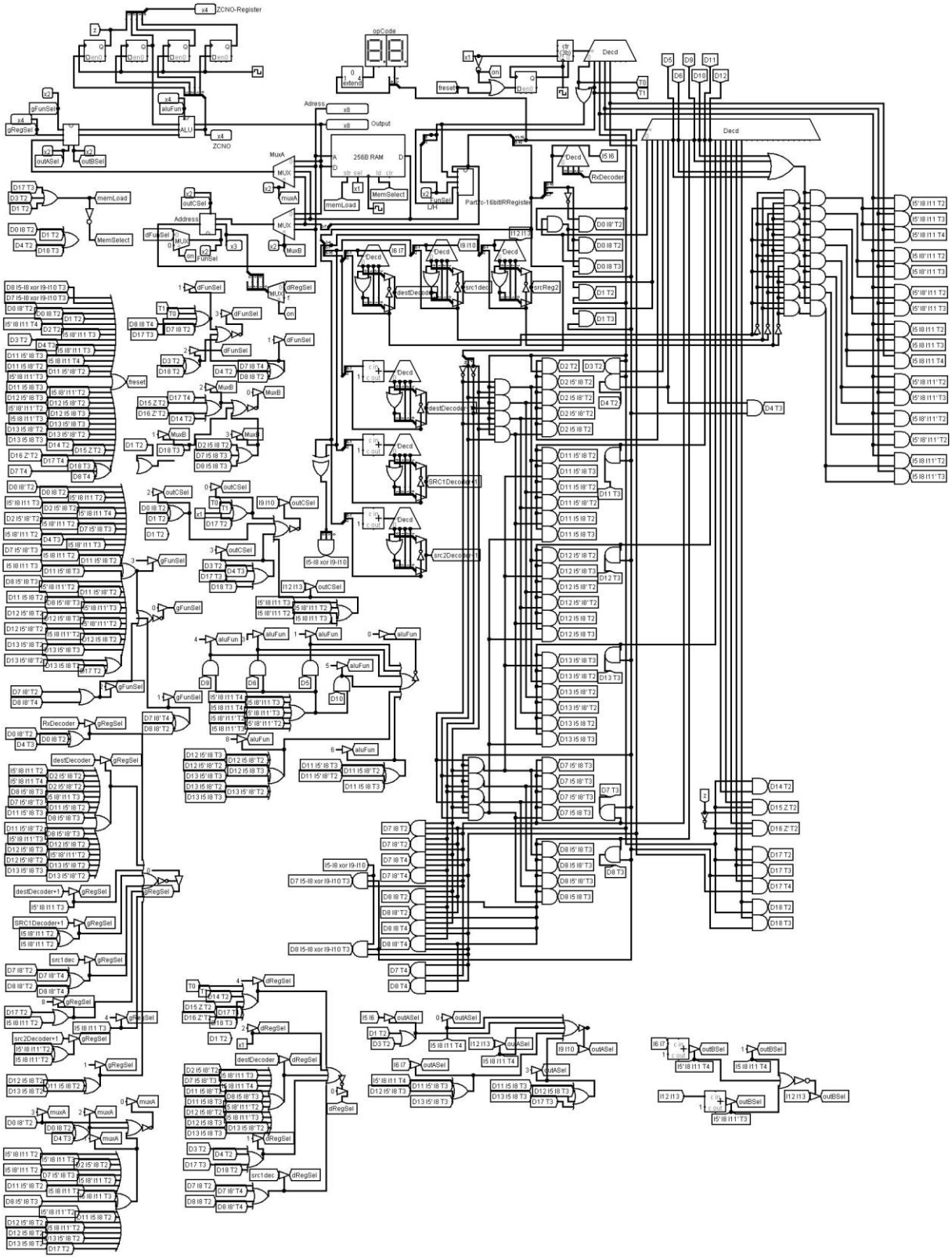
DESTREG/SRCREG1/SRCREG2	REGISTER
000	R0
001	R1
010	R2
011	R3
100	PC
101	PC
110	AR
111	SP

Table 3: Addressing modes

ADDRESSING MODE	MODE	SYMB	Value
0	Immediate	IM	ADDRESS Field
1	Direct	D	M[AR]

The directions and corresponding signals are given in separate excel files since they were too big to fit in this report, see *Sinyaller.xlsx* and *Komutlar ve Sinyaller.xlsx* for detailed information.

The circuit implemented is given on the next page; some parts of the circuit were realized as parts of the previous projects. Check the circuit on Logisim for detailed insight.


ZCNO-Register
opCode
ALU
ZCNO
Adress
Output
256B RAM
MuxA
MuxB
memLoad
MemSelect
Decd
RxDecoder
destDecoder
SRC1Decoder
src2Decoder
gFunSel
gRegSel
aluFun
outASel
outBSel
outCSel
dFunSel
dRegSel
muxA
muxB
freset

The test case and the corresponding machine code are given as follows:

				OpCode1						
			OpCode	RegSel(R	a mode	adress				
			I0-I4	I5-I6	I7	I7-I0				
			15-11	10-9	8	7-0				
				OpCode2						
			OpCode	DestReg	SrcReg1	SrcReg2				
			I0-I4	I5-I7	I8-I10	I11-I13	I14-I15			
			15-11	10-8	7-5	4-2	2-0			
		ORG 0x20								
0X00		BRA 0X20	01110	00	0	00100000		0111000000100000	7020	
0X20		LD R0 IM 0x05	00000	00	0	00000101		0000000000000101	0005	
0X22		LD R1 IM 0x00	00000	01	0	00000000		0000001000000000	0200	
0X24		0xA0	00000	10	0	10100000		0000010010100000	04A0	
0X26		MOV AR R2	00010	110	010	000	00	0001011001000000	1640	
0X28	LABEL:	LD R2 D	00000	10	1	00000000		0000010100000000	0500	
0X2A		INC AR AR	01000	110	110	000	00	0100011011000000	46C0	
0X2C		ADD R1 R1 R2	00101	001	001	010	00	0010100100101000	2928	
0X2E		DEC R0 R0	00111	000	000	000	00	0011100000000000	3800	
0X30		BNE IM LABEL	10000	00	0	00101000		1000000000101000	8028	
0X31		INC AR AR	01000	110	110	000	00	0100011011000000	46C0	
0X32		ST R1 D	00001	01	1	00000000		0000101100000000	0B00	
0X33										

3. CONCLUSIONS

In this project, we implemented a full hardware-control unit that has its own fetch-decode cycles and is able to perform various operations predefined in its instruction set.