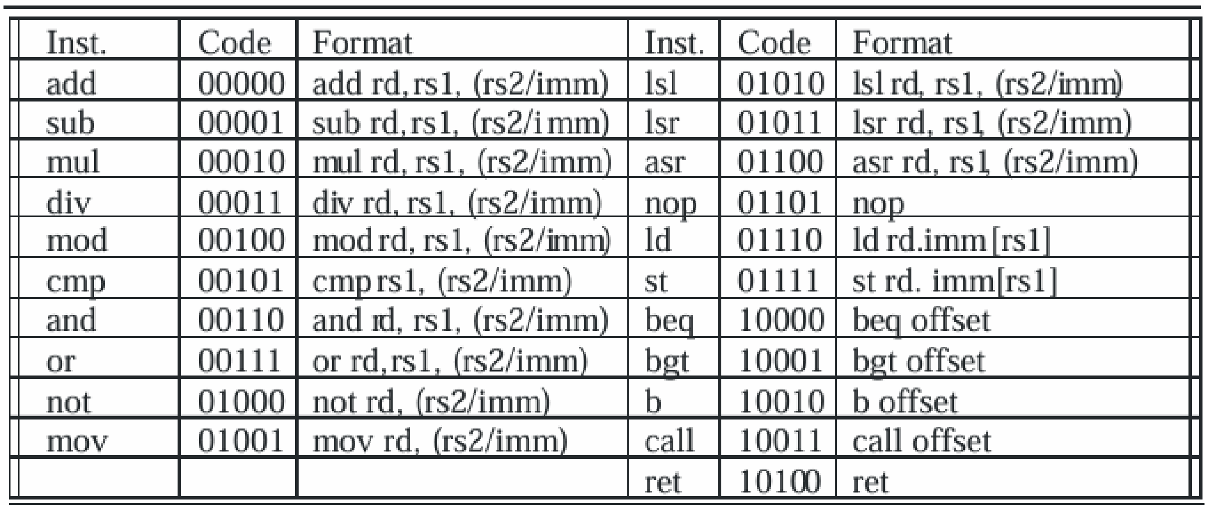
Single Cycle Processor

This processor uses a RISC Instruction Set Architecture (ISA) named ***SimpleRisc***. This ISA allows 32 bit of instructions and contains ***21 operators*** hence ***5-bit opcode*** for each operand. It uses sixteen 32-bit registers out of which two are used for special purpose like register-14 is used as Program Counter register and register-15 is used as return address register.

Following are the operators, opcode, and format used in this ISA-

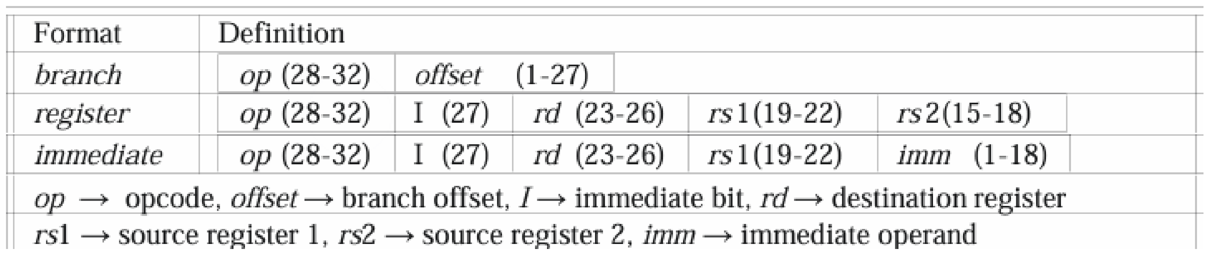


Inst.	Code	Format	Inst.	Code	Format
add	00000	add rd, rs1, (rs2/imm)	lsl	01010	lsl rd, rs1, (rs2/imm)
sub	00001	sub rd, rs1, (rs2/imm)	lsr	01011	lsr rd, rs1, (rs2/imm)
mul	00010	mul rd, rs1, (rs2/imm)	asr	01100	asr rd, rs1, (rs2/imm)
div	00011	div rd, rs1, (rs2/imm)	nop	01101	nop
mod	00100	mod rd, rs1, (rs2/imm)	ld	01110	ld rd.imm[rs1]
cmp	00101	cmp rs1, (rs2/imm)	st	01111	st rd. imm[rs1]
and	00110	and rd, rs1, (rs2/imm)	beq	10000	beq offset
or	00111	or rd, rs1, (rs2/imm)	bgt	10001	bgt offset
not	01000	not rd, (rs2/imm)	b	10010	b offset
mov	01001	mov rd, (rs2/imm)	call	10011	call offset
			ret	10100	ret

For encoding the instructions these are categorized in mainly 4 formats –

1) ***Branch instruction format*** – In this category we encode nop, ret, call, b, beq and bgt instructions.
2) ***Register instruction format*** – All ALU instruction are being encoded in this format only.
3) ***Immediate instruction format*** – ALU and ld/st instructions are encoded in this format basically this format is used to use constants.

Following is the summary of the instruction formats-



Format	Definition				
branch	*op* (28-32)	*offset* (1-27)			
register	*op* (28-32)	I (27)	*rd* (23-26)	*rs*1(19-22)	*rs*2(15-18)
immediate	*op* (28-32)	I (27)	*rd* (23-26)	*rs*1(19-22)	*imm* (1-18)

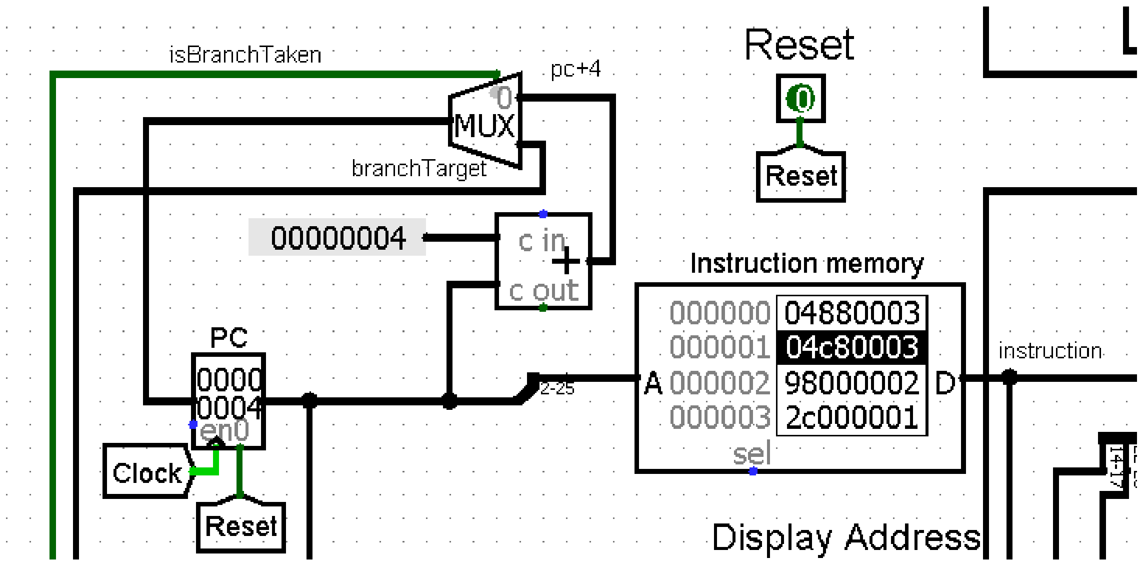
op → opcode, *offset* → branch offset, *I* → immediate bit, *rd* → destination register
*rs*1 → source register 1, *rs*2 → source register 2, *imm* → immediate operand

Processor Design –

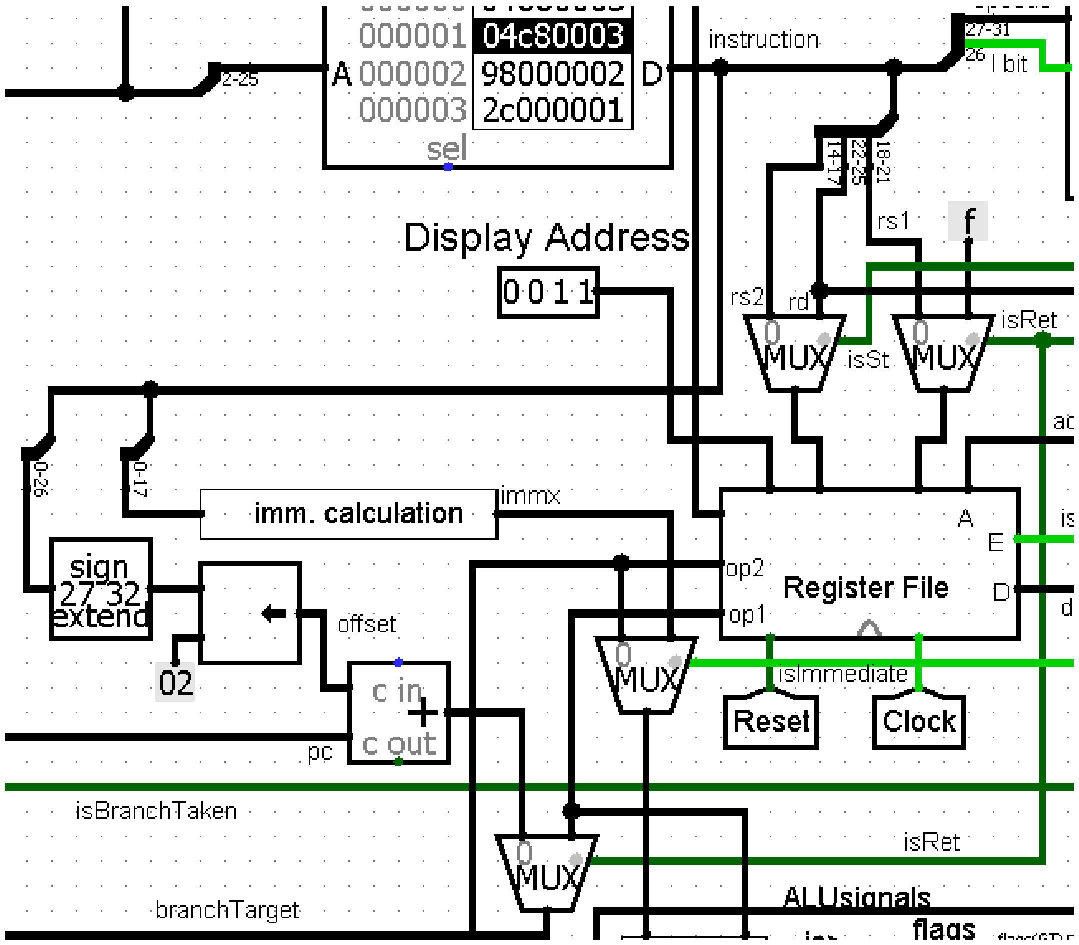
This processor is divided into following 4 stages –

1. ***Instruction Fetch –*** In this stage of the processor we fetch the instruction from the instruction memory via address in PC register. The address stored in the PC is the result of either the adder that adds four in its content or it is the address comes from

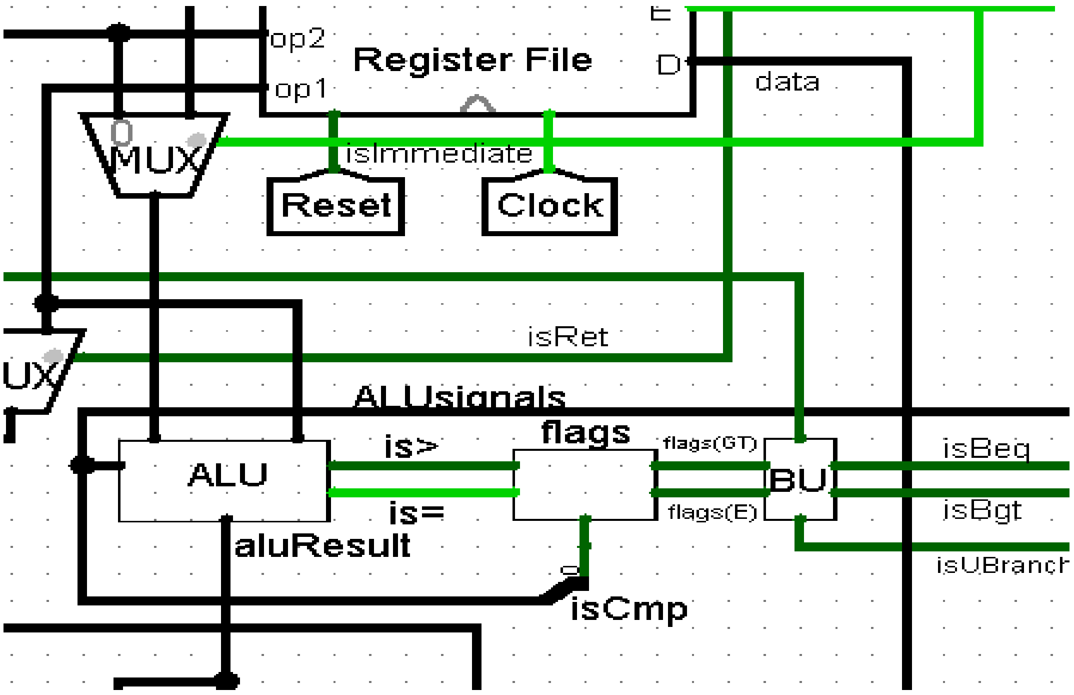
the branch instruction. The following part of the processor represent the Instruction Fetch unit –



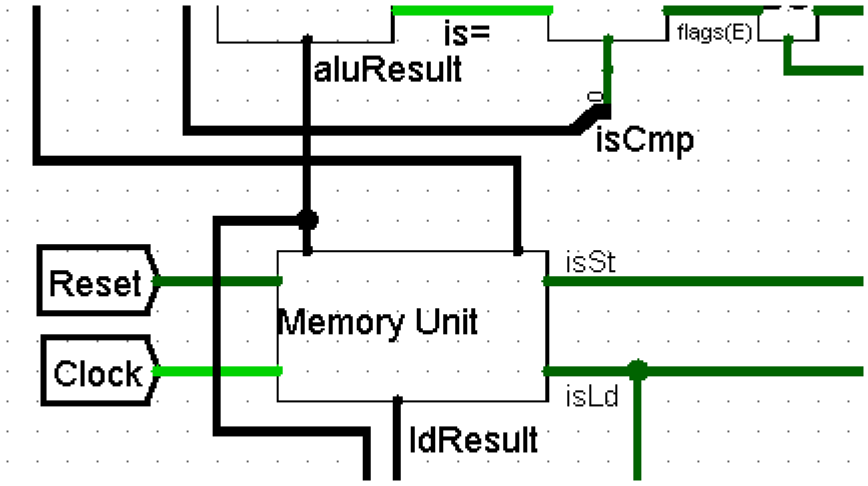
2. ***Operand Fetch –*** This stage of processor fetches the value of operands form the register file through the addresses in the instruction or the return address register. We also calculate the value of the immediate and branch target in based on the value of offset given in the instruction. And then we choose the target that goes to PC based on the isRet signal which state if the we have to move on return address. Following part of the processor represents the OF unit –



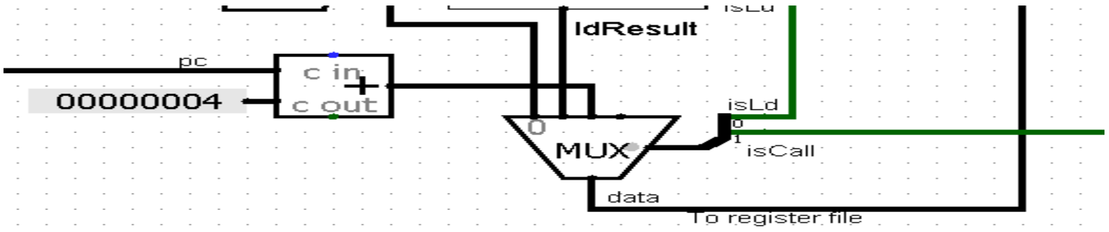
3. ***Execute Stage –*** This unit of the processor circuit contains ALU, Branch unit, and flags register. ALU is used to perform all arithmetic operations (add, sub, mul, div, cmp, mod) and all logical operations (and, or, not). Branch unit is used for computing the branch conditions(beq, bgt) and flags register store the value of the cmp instruction.



4. ***Memory Access Stage –*** This unit is used to interface with the memory it either load the value from the memory or store the value to the memory based on load or store instructions. Following part of the processor circuit is MA stage –

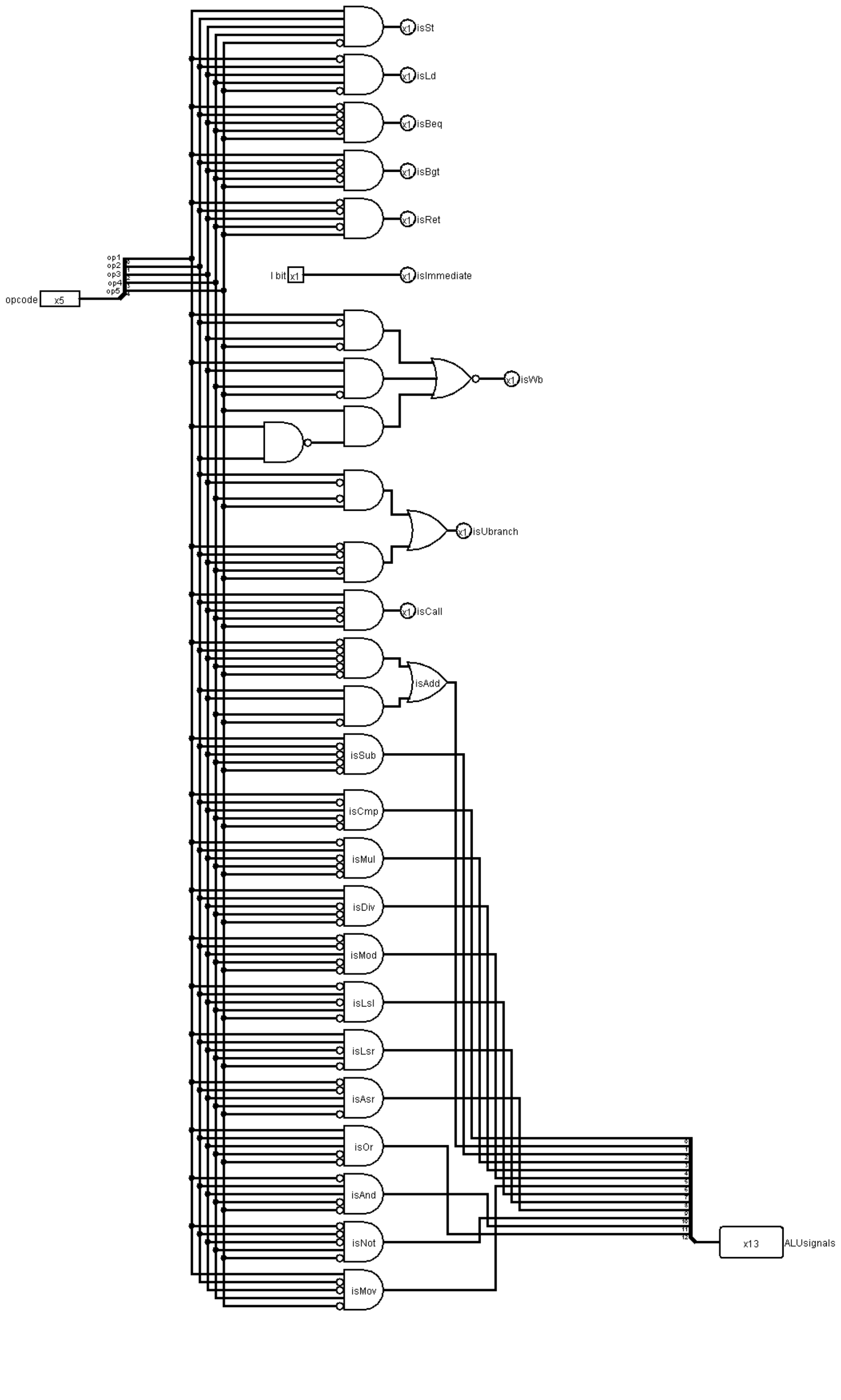


5. ***Register Write Stage –*** This stage writes the required value in the register file like the load result, return address, or aluResult. Following part of the circuit represent the RW stage of the processor –

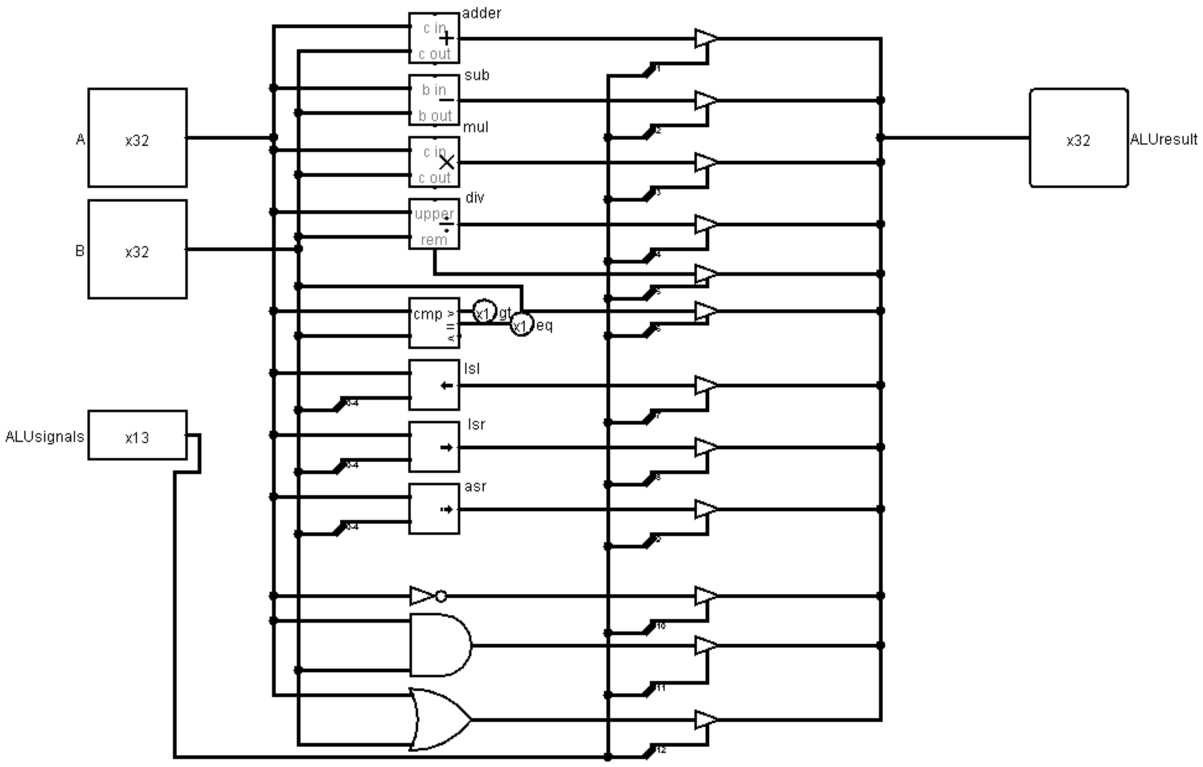


Design of the components used in the processors – Many components like control unit, alu, memory unit, branch unit, flags, immediate calculation and register files are also being designed in the same project. They use basic combinational design logic for their operations. Following are brief detail of their design –

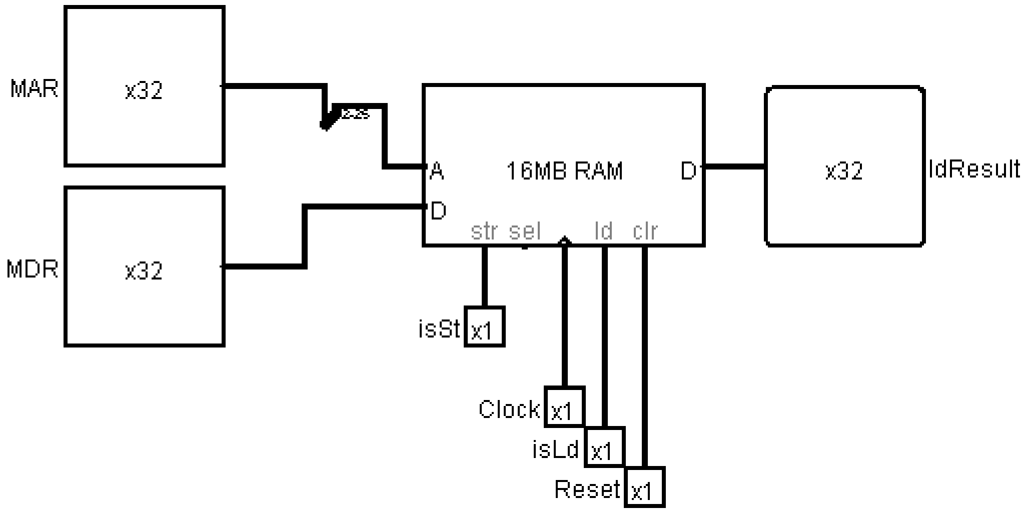
a. ***Control Unit –*** In our processor we used hardwired control unit that generates control signals based on the opcode values. Following is the image of it -



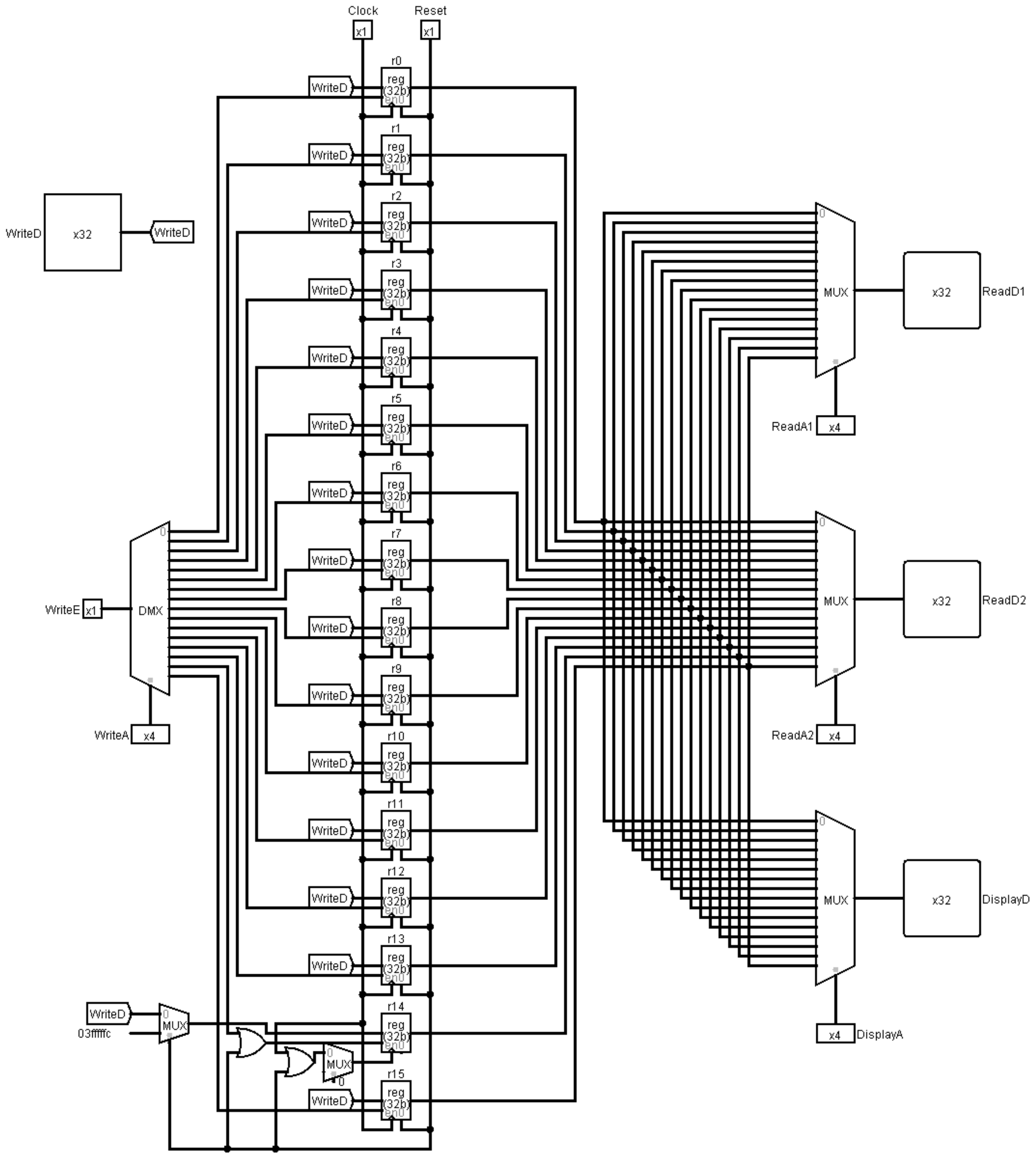
b. ***ALU –*** ALU of this processor can perform arithmetic operations like addition, subtraction, multiplication, division, comparison and remainder calculation alco, logical operations like and, or and not. Following is the circuit of the ALU used in this processor -



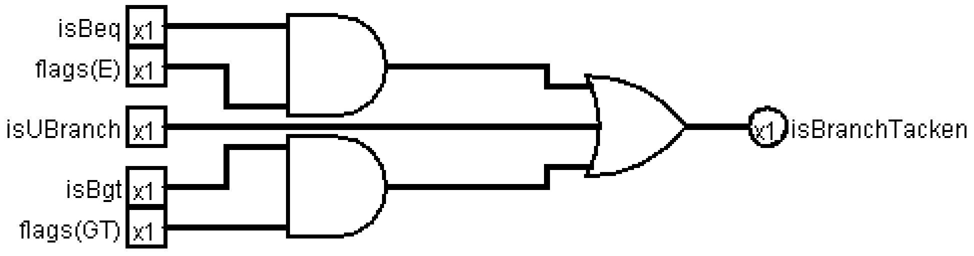
c. ***Memory Unit –*** The memory unit interacts with the data memory and perform store and load operations following is the image of it –



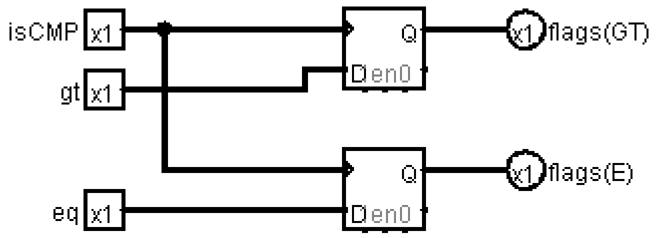
d. ***Register File –*** Register file is the set of 16-Register that are accessed by the instructions, save the results and used to load the values from memory. Following is the image of the Register file –



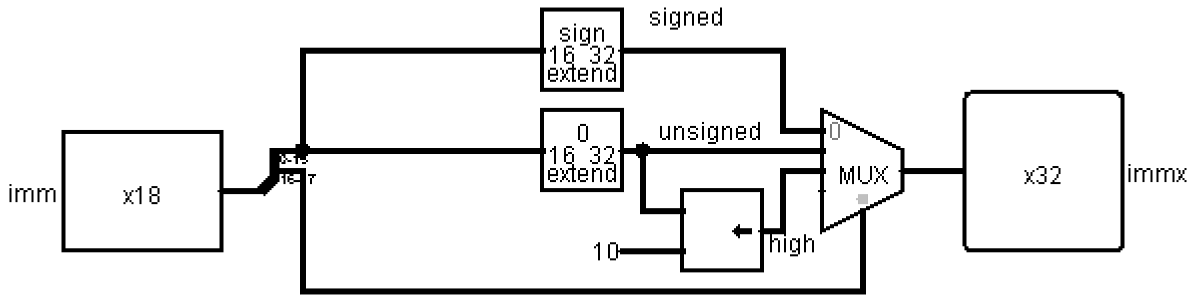
e. ***Branch Unit –*** This unit is used to decide whether the branch action is to be taken or not based on the beq, bgt operations and compare results. Following is the image of the circuit -



f. ***Flags –*** This unit is used to set the flags register value based on the comparison results of ALU. Following is the circuit of the Flags unit –



g. ***Immediate Calculation –*** This unit is used to expand the immediate based on their use like unsigned, signed or shifting them to the MSB. Following is the circuit diagram of this unit –


imm
x18
sign
16 32
extend
signed
0
16 32
extend
unsigned
10
high
MUX
0
1
x32
immx