CS 317: Computer Organization
Project 1: Proposed Ops and Opcodes

1 Overview

Your assignment will be to design, describe, and build a simple processor *similar to* that discussed in Chapter 5 of your textbook. We will all be using the same datapath, shown in Figure 5.18 of your textbook and at the end of this packed and variations on the IR fields shown below (and in Figure 5.12 of your text).

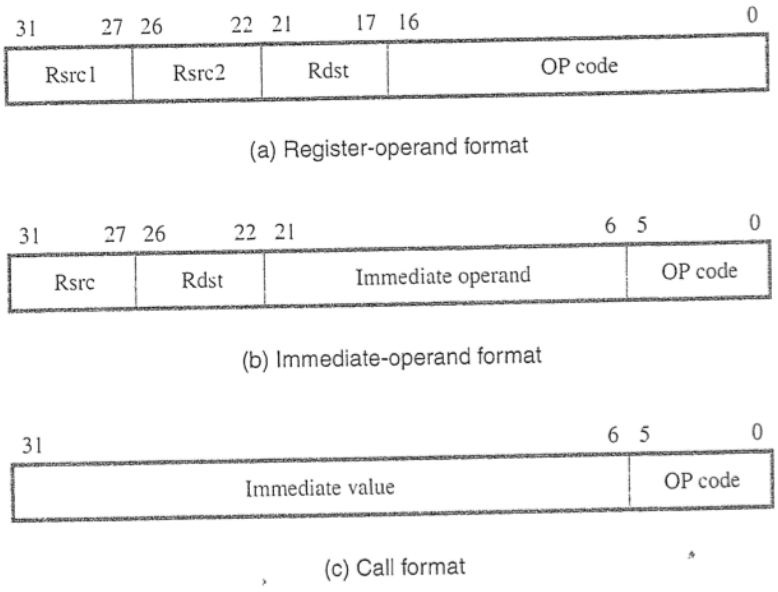

Figure 5.12 Instruction encoding.

All processor busses and registers should be 32 bits wide, and you should have 32 (nominally) general purpose registers, numbered R0-R31. All of the registers shown in the datapath are also 32-bit registers. This implies that you will need five bits to specify each of the register fields, as shown above. We will use variations of the textbook formats when implementing our processor. And you will support following functions and the corresponding instruction word formats, subject to discussion in class on Friday, October 17th.

I've sorted these by which format in Figure 5.12 is most relevant, and given on the first line the opcode, a mnemonic to refer to the instruction by, and an optional phrase the mnemonic stands for if I don't think it's obvious. This is followed by a brief description.

Variations on Format (a)

XXXX_XXXXXXX_111111 NOP No OPeration
Do nothing (but take all five cycles to do it) Note that this is any instruction with the 6 least significant bits equal to 111111, so this pattern can not be used for formats (b) or (c) either.

0000_0000010 _000000 ADD
The contents of Rsrc1 and Rsrc2 are added, the result is stored in Rdst

0000_0000100 _000000 SUB
The contents of Rsrc2 are subtracted from the contents of Rsrc1 and the result is stored in Rdst

0000_0001100 _000000 COMP
The bitwise (one's) complement of the contents of Rsrc1 are placed in Rdst

0000_0001010 _000000 NEG
The two's complement of the contents of Rsrc1 are placed in Rdst

0000_0001000 _000000 AND
The contents of the two source registers are logically ANDed together (bitwise AND) and the result is placed in Rdst

0000_0001001 _000000 OR
The contents of the two source registers are logically ORed together (bitwise OR) and the result is placed in Rdst

0000_0001011 _000000 XOR The contents of the two source registers are logically XORed together (bitwise XOR) and the result is placed in Rdst

0000_0010010 _000000 ASL Arithmetic Shift Left (shift one bit position only)
This is a 33 bit shift of the contents of Rsrc1 with the left-most bit of Rsrc1 going to the Carry flag and a 0 being shifted into the right-most bit position of Rdst

0000_0010001 _000000 ASR Arithmetic Shift Left (shift one bit position only)
This is a 33 bit shift of Rsrc1 with the right-most bit of Rsrc1 going to the Carry flag and a sign extension used to fill left-most bit position of Rdst

0000_0011010 _000000 LSL Logical Shift Left (shift one bit position only)
This is a 33 bit shift of Rsrc1 with the left-most bit of Rsrc1 going to the Carry flag and a 0 being shifted into the right-most bit position of Rdst

0000_0010010 _000000 LSR Logical Shift Left (shift one bit position only)
This is a 33 bit shift of Rsrc1 with the right-most bit of Rsrc1 going to the Carry flag and a 0 being shifted into the left-most bit position of Rdst

0000_0011010 _000000 ROL Rotate Left (by one bit position)
Affects carry bit in the same manner as the shifts

0000_0011001 _000000 ROR Rotate Right (by one bit position)
Affects carry bit in the same manner as the shifts

0000_0100000 _000000 MOVE (Copy)
The contents of Rsrc1 are copied into Rdst

0000_0100001 _000000 LBI Load Base with Index
The unsigned contents of the two source registers are added to create the EA. Rdst is then loaded from memory location EA

0000_0100010 _000000 LDRi Load Register Indirect
Rsrc2 contains a pointer to the value to be copied into Rdst

0000_1000000 _000000 JMP Jump
Place the contents of Rsrc1 into the PC

0000_1000001 _000000 JSR Jump to Subroutine
Address of subroutine in Register Rsrc1, store return address in LINK register, which is always R30.

0000_1000011 _000000 RTS Return from Subroutine
Rsrc1 contains the register number for the link register (R30).

Variations on Format (b)

000010 ADD #
The immediate value is sign extended and added to the contents of Rsrc. The result is stored in Rdst

000100 SUB #
The immediate value is sign extended and subtracted from the contents of Rsrc. The result is stored in Rdst

001000 AND #
The immediate value is padded with zeros on the left and ANDed with the contents of Rsrc. The result is place in Rdst

001001 OR #
The immediate value is padded with zeros on the left and ORed with the contents of Rsrc. The result is place in Rdst

001011 XOR #

The immediate value is padded with zeros on the left and XORed with the contents of Rsrc. The result is place in Rdst

001100 BEQ Branch if EQual

If the contents of the two registers are equal, add the 2's complement immediate value to the PC

001010 BNE Branch if Not Equal

If the contents of the two registers are not equal, add the 2's complement immediate value to the PC

001111 BLT Branch if Less Than

If the unsigned contents of Rsrc are less than the contents of Rdst, add the 2's complement immediate value to the PC

100000 LDA Load Absolute

The immediate value is zero-filled to the left and used as an address. Rdst is then loaded from this address. This requires the adder in Figure 5.10 to be able to just pass the immediate value through (without adding to the PC), which requires an additional control line that is not implied by Figure 5.10

010000 STA STore Absolute

The immediate value is zero-filled to the left and used as an address. Rdst (yes, Rdst!) is then stored to this address. Requires modifications similar to the LDA instruction

100001 LDIX LoaD IndeXed

The unsigned immediate value is added to the contents of Rsrc to obtain the EA. Rdst is then loaded from the memory location EA

010001 STIX STore IndeXed

The unsigned immediate value is added to the contents of Rsrc to obtain the EA. Rdst is then stored to the memory location EA

100010 LD # Load immediate

The value in the immediate field is sign extended and placed in the Rdst.

100011 LDU Rx #num Load unsigned immediate

The value in the immediate field is padded with zeros to the left and placed in Rdst

Variations on Format (c)

110000 BRA Branch

Add the 2's complement immediate value to the PC.

```
110001 BSR          Branch to SubRoutine
    Add the 2's complement immediate value to the PC and store return address in
    the LINK register, which is always R30.
```

We will discuss these operations and opcodes in class on Friday prior to our discussion on memory.

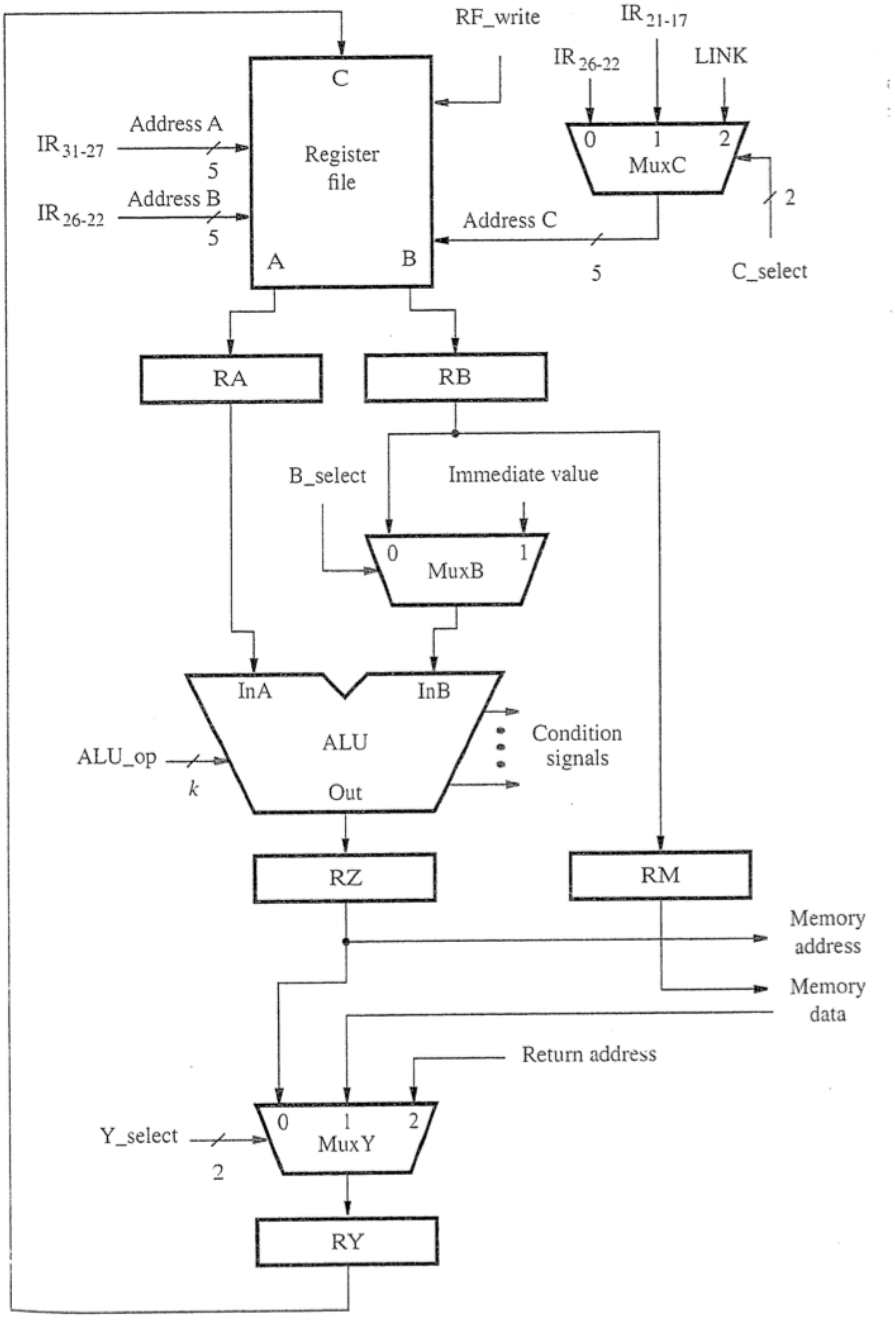

Figure 5.18 Control signals for the datapath.