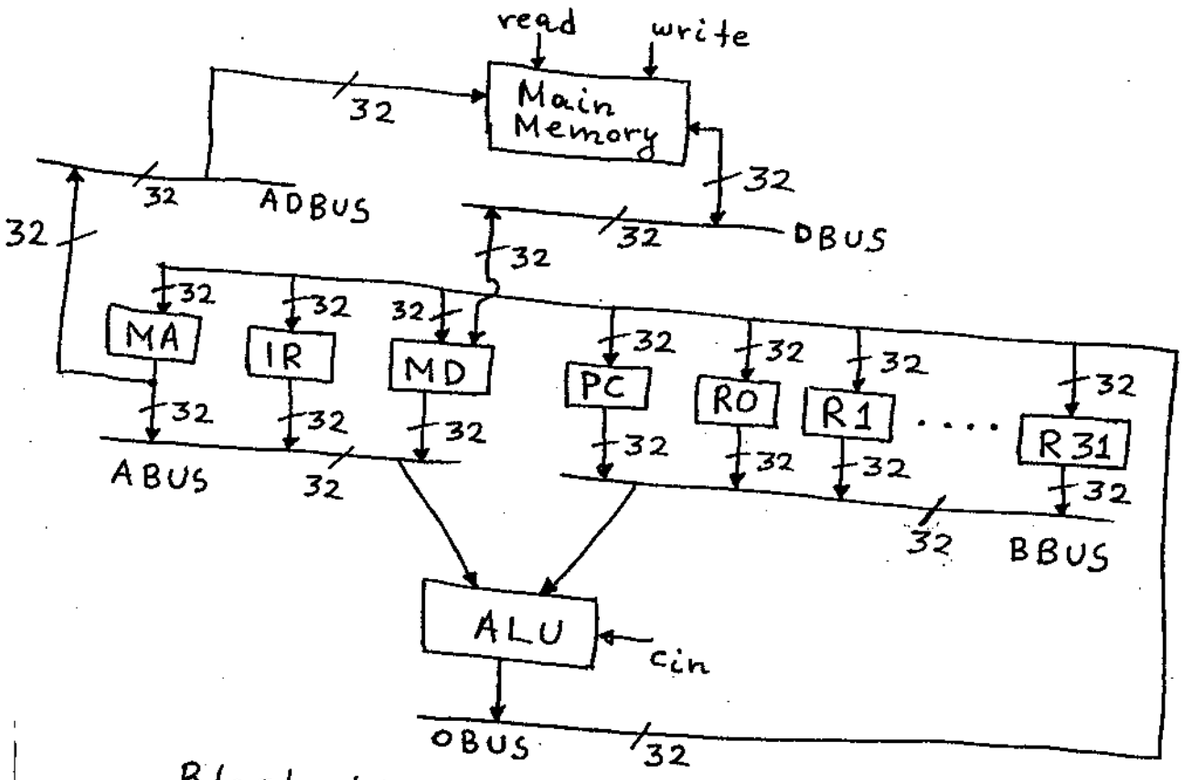
EE 3755

Handout entitled:

Second example computer

Another computer named "second example computer" is presented here.



Block diagram of the computer.

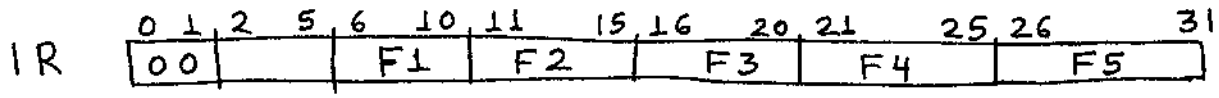
- The computer has 32 32-bit general-purpose registers R0; R1; ...; R31. The register R29 is used as the stack pointer.
- All the instructions of the computer are 32-bit instructions.

- The [redacted] computer is a load-and-store architecture. This means that only the load and store instructions reference memory, while all instructions that operate on data are register-to-register instructions. The load instruction results in loading a register with memory-data while the store instruction results in storing a register in memory.
- Flags: cff; zff; nff; vff.
- Instruction Formats: There are four instruction formats. These are: Format A, Format B, Format C and Format D. The instruction field IR[0:1] dictates the format as shown below:

IR[0:1]	Format
00	A
01	B
10	C
11	D

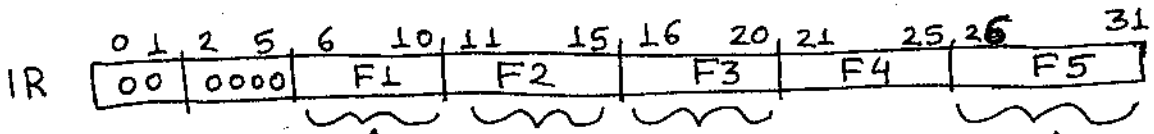
The instructions of the [redacted] computer are next explained in details.

1. Format A Instructions:



0000 = Arithmetic Instruction
0001 = Logic Instruction
0010 = Shift/rotate Instruction
0011 = Jump register instruction
0100 = Jump subroutine register instr.
0101 = Special purpose instruction

1.1. Format A arithmetic instructions:



00000 = reg. R0
00001 = reg. R1
00010 = reg. R2
00011 = reg. R3
⋮
11111 = reg. 31

000000 = ADD (add)
000001 = ADC (add with carry)
000010 = SUB (subtract)
000011 = SBC (subtr. with carry).
000100 = CMP (arith compare; same as SUB except that no target is affected).
⋮

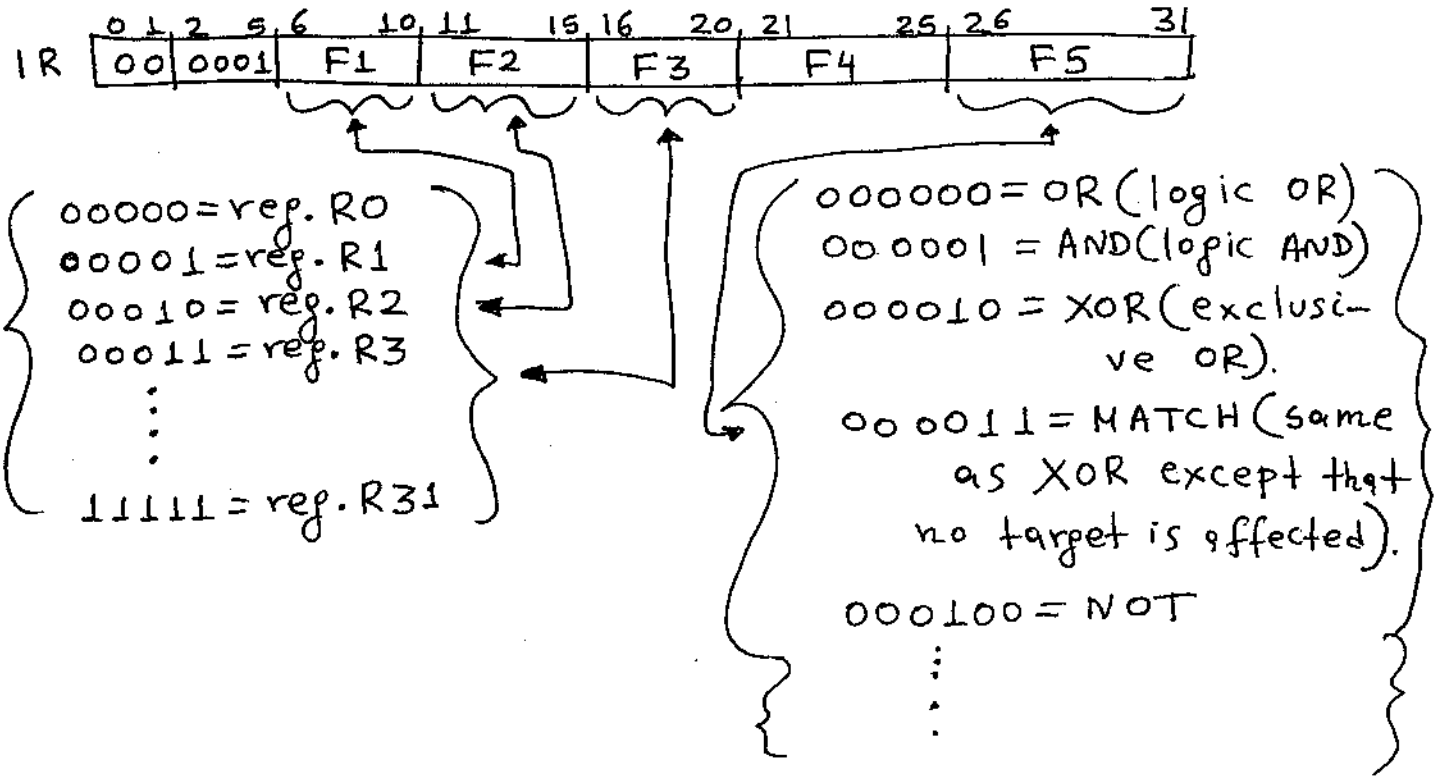
1.1.1. Meaning of format A arithmetic instructions:

register specified by F3 ← (reg. specif. by F1) ⊙ (reg. specif. by F2)

where ⊙ = ADD ; ADC ; SUB ; SBC ; CMP ; ...

- All four flags cff zff nff vff are affected by the format A arith. instructions.

1.2. Format A logic instructions:



1.2.1. Meaning of format A logic instructions:

register specified by F3 ← (reg. spec. by F1) ⊙ (regist. spec. by F2).

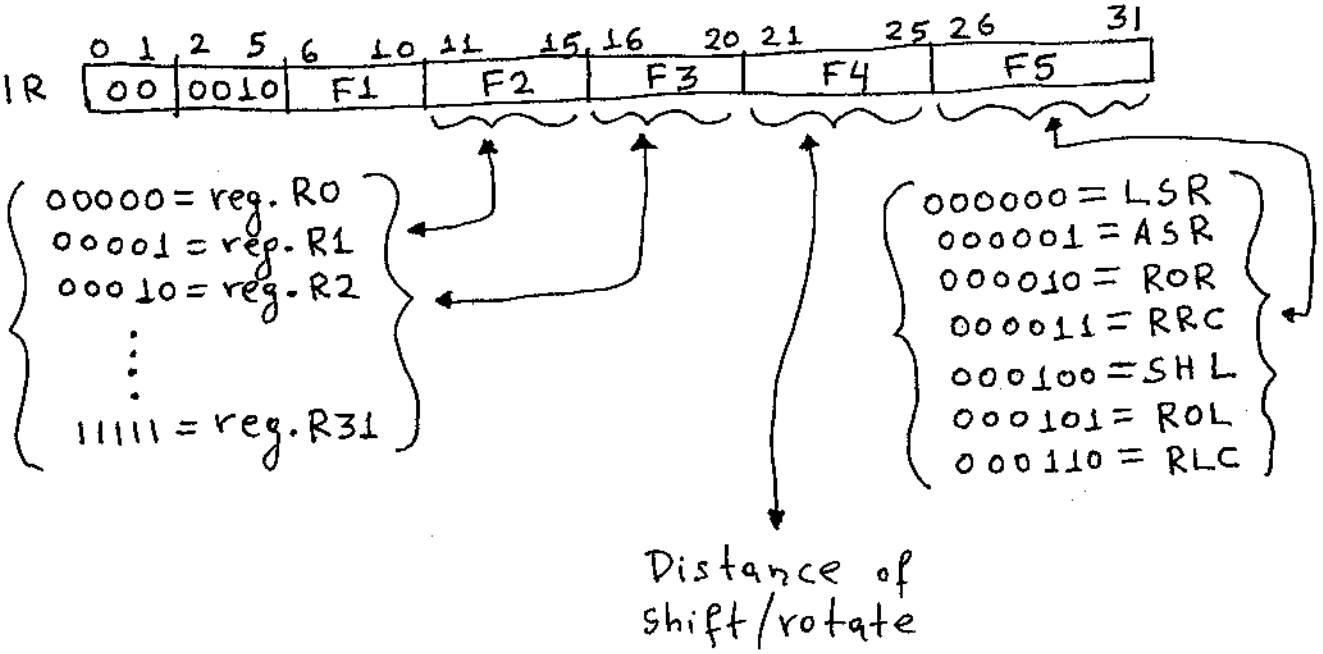
where ⊙ = OR; AND; XOR; MATCH; ....

The meaning of the instruction NOT is

register specified by F3 ← ($\overline{\text{reg. spec. by F2}}$)

• Flag zff affected by the logic instructions.

1.3. Format A shift/rotate instructions

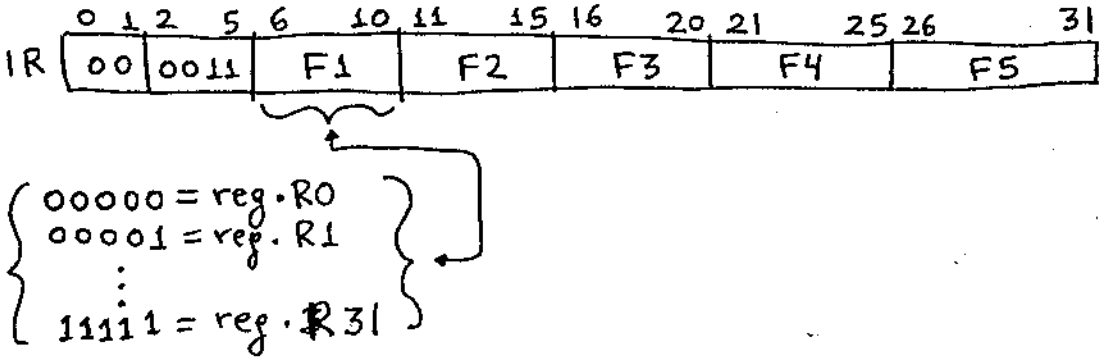


1.3.1. Meaning of format A shift/rotate instructions:

shift/rotate register specified by F2 and store result in register specified by F3

For the shift/rotate instructions, the various shift/rotate transactions are the same as the ones for the "First example computer"; (see your handouts etc). For example, LSR is logical shift right, ASR is arithmetic shift right, ROR is rotate right, RRC is rotate right with carry, SHL is shift left, ROL is rotate left, RLC is rotate left with carry. (see appropriate handout on "First example computer"). The cff flag is affected by all the above except the ROR and ROL; (see appropriate handouts etc...).

1.4. Format A jump register instructions

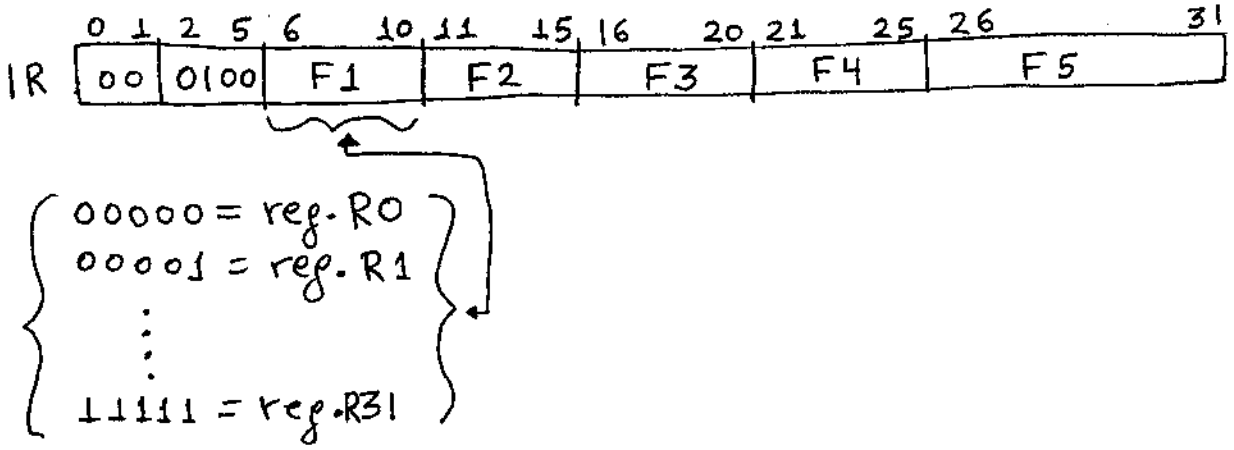


1.4.1. Meaning of format A jump register instr.

Jump to address which is found in the register specified by the field F1. In other words, the next instruction will be fetched

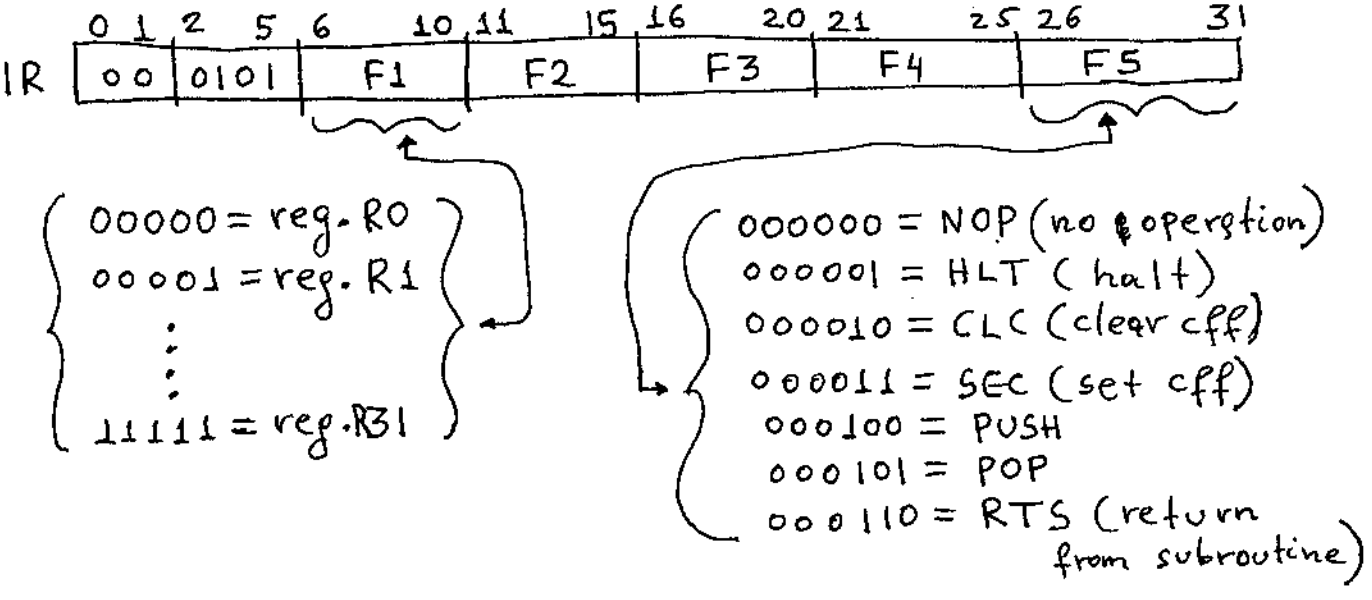
from memory location with address being the contents of the register specified by the field F1. No flags are affected.

1.5 Format A jump subroutine register instructions:



The meaning here is jump to subroutine starting at address found in the register specified by the field F1. No flags are affected.

1.6 Format A special purpose instructions:



The meaning of the instructions NOP, HLT, CLC, SEC, RTS has already been explained. Regarding the instructions PUSH and POP we have:

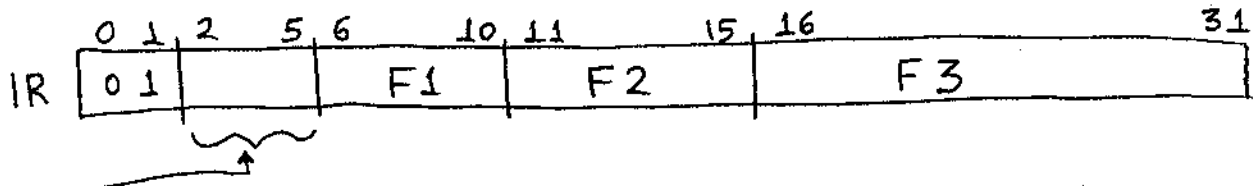
Meaning of PUSH :

stack ← contents of register specified by field F1

Meaning of POP:

regist. specified by F1 ← contents of top of stack

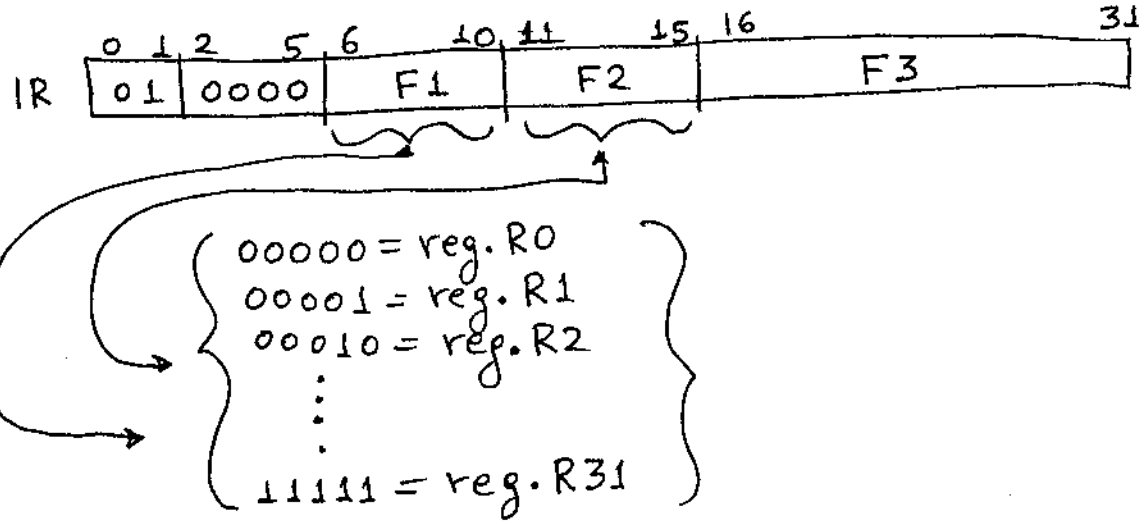
2. Format B Instructions



0000 = Load Instruction
0001 = Store Instruction
0010 = Register Transfer Instruction
0011
0100
0101
0110
0111
= Arithmetic Instructions
1000 = Load Lower Immediate
1001 = Load Upper Immediate

For these Format B instructions, the 5-bit fields F1 = IR[6:10] and F2 = IR[11:15] specify any of the 32 registers R0 R1 .... R31. The 16-bit field F3 = IR[16:31] specifies either address or immediate data. Details of the various Format B instructions follow:

2.1. Format B load instruction:



The meaning of the load instruction is:

register specified by field F2 ← Memory data

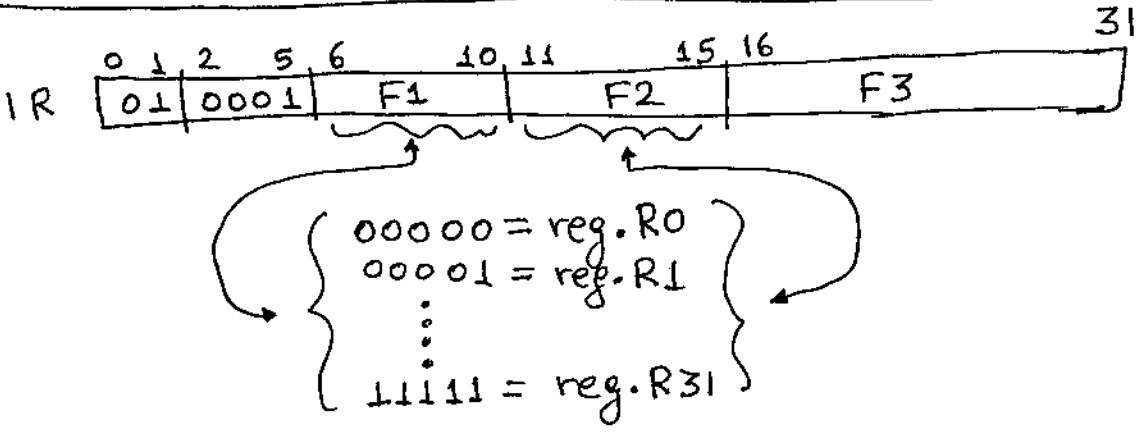
The memory data are fetched from address

address = F3 + (contents of register specified by field F1)

Here, the 16-bit field F3 = IR[16:31] participates in the calculation of the memory-address.

The flags zff and nff are affected

2.2. Format B store instruction:



The meaning of the store instruction is:

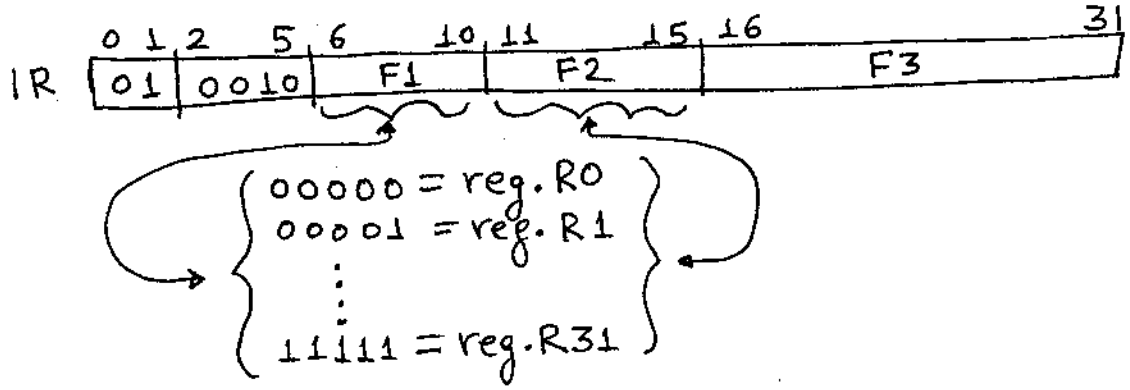
> Memory ← (register specified by field F2)

The register specified by field F2 = IR[11:15] is stored in memory location with address being

> address = F3 + (contents of register specified by field F1)

Recall that the field F3 is IR[16:31]; (the 16-bit rightmost field of the store instruction).
No flags are affected.

2.3. Format B register transfer instruction:

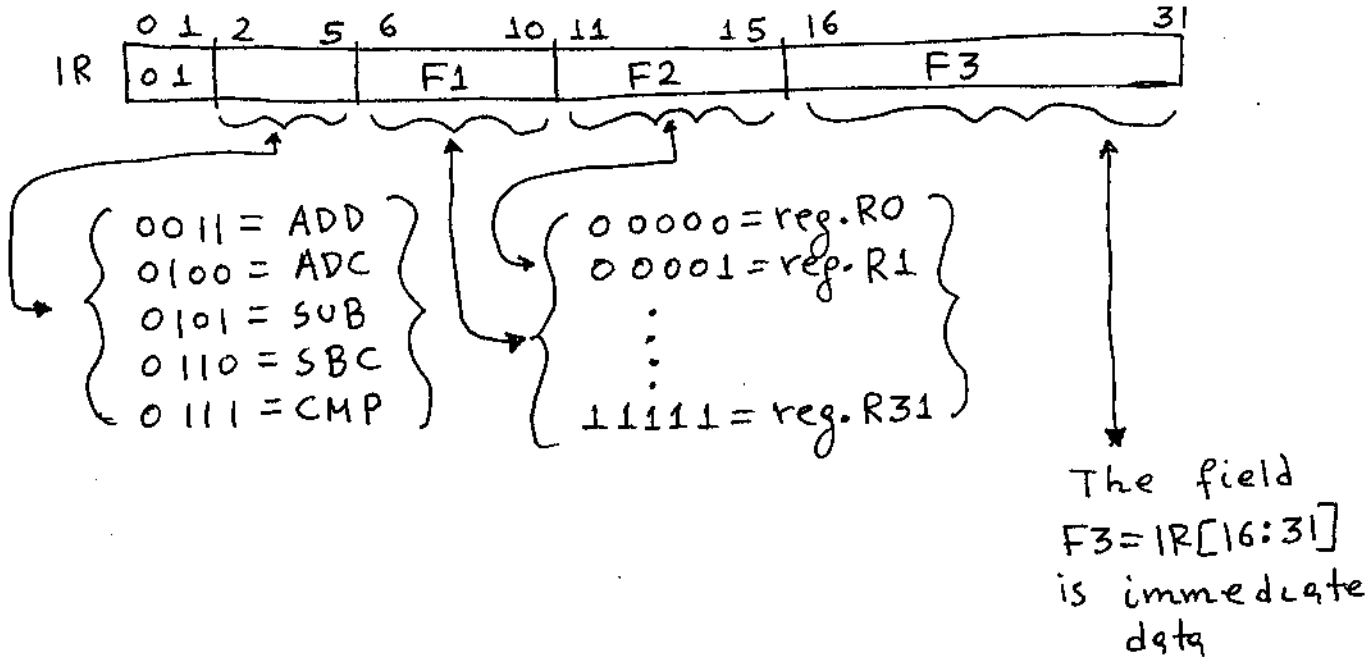


The meaning of the register transfer instr. is:

> register specified by field F2 ← (regist. specified by field F1)

The flags zff and nff are affected.

2.4 Format B arithmetic instructions



The meaning of these arithmetic instr. is:

register specified by field F2 ← (reg. specified by F1) ⊙ (immediate data)

where ⊙ = ADD; ADC; SUB; SBC; CMP.

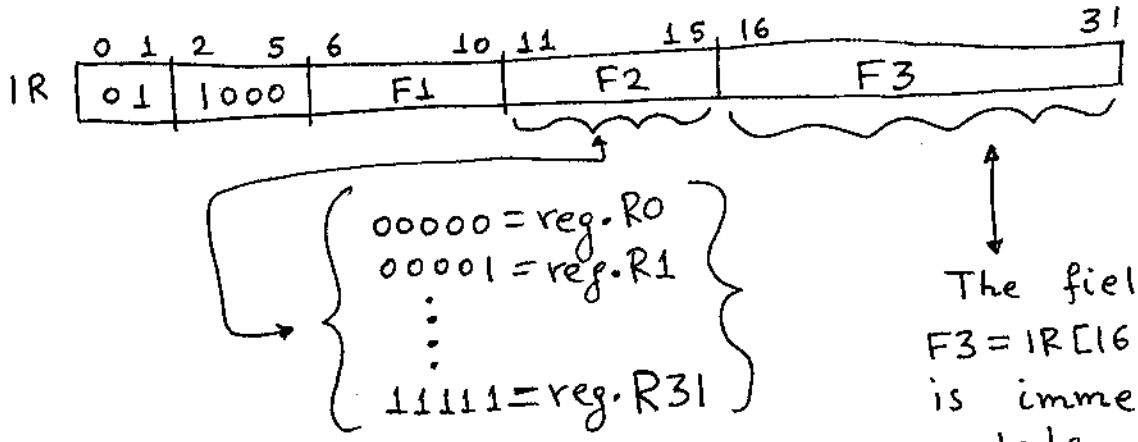
The five arithmetic instructions are the well known ADD (add); ADC (add with carry); SUB (subtract); SBC (subtract with carry); CMP (arithmetic compare).

The above arithmetic instructions operate between register data and immediate data. The immediate data are provided by the field F3 = IR[16:31]

of the instruction itself and must be sign extended.

All four flags cff zff nff vff are affected.

2.5 Load Lower Immediate Instruction

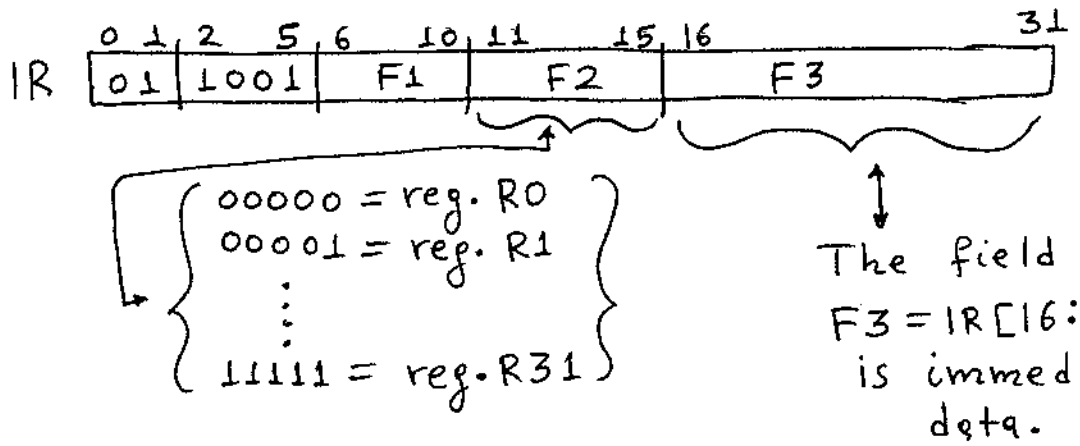


The meaning of this instruction is:

16-bit rightmost part of reg. specified by F2 ← immediate data

In simple words, this instruction loads immediate 16-bit data to the 16-bit lower (rightmost) part of the register specified by the instruction field F2 = IR[11:15]. The 16-bit leftmost (upper) part of the register remains unaffected. The 16-bit immediate data are provided by the field F3 = IR[16:31] of the instruction.

2.6 Load Upper Immediate Instruction



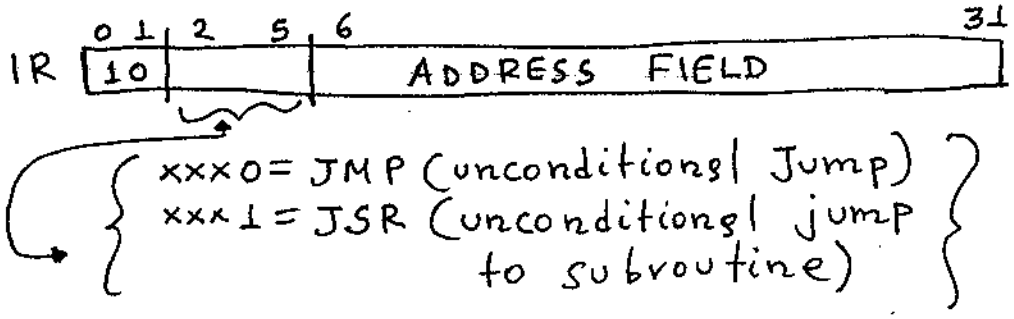
The meaning of this instruction is:

16-bit leftmost part of reg. specified by F2 ← immediate data

Clearly, this instruction loads 16-bit immediate data to the 16-bit upper (leftmost) part of the register specified by field F2. The 16-bit rightmost (lower) part of the register remains unaffected. The 16-bit immediate data are the field F3 = IR[16:31] of the instruction.

It is clear that the load upper immediate and load lower immediate can result in loading any one of the general-purpose registers R0 .... R31 with any 32-bit immediate data.

3. Format C Instructions:



These two format C instructions are the JMP (unconditional jump) and the JSR (unconditional jump to subroutine).

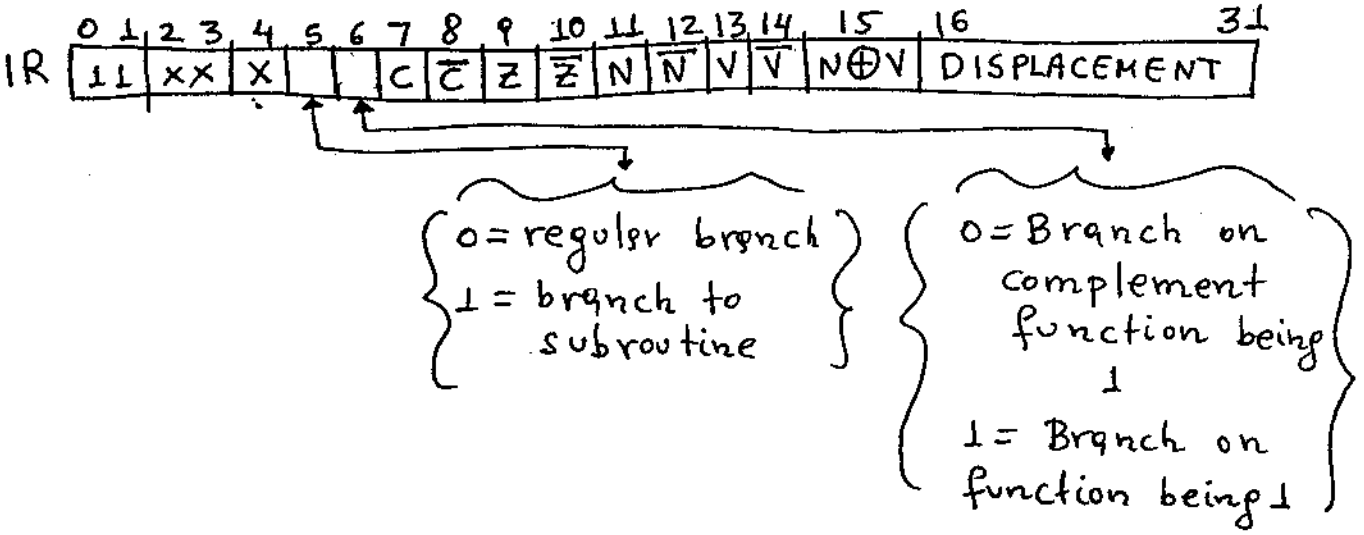
The meaning of JMP is to jump to address = ADDRESS FIELD = IR[6:31] (see instruction above).

Regarding JSR, the meaning is jump to subroutine starting at address equal to IR[6:31] = ADDRESS FIELD (see instruction above).

No flags are affected.

4. Format D Instructions

The format D instructions are the same as the Branch instructions of the "First example computer"; (see appropriate handout).



* Review the handout entitled :
"First example computer : Part (3)"

for details on the branch instruction etc ... etc.

• Note : The instructions of the above presented computer ("The second example computer") have few similarities with the instructions of the "First example computer".

The "Second example computer" has many similarities with a real computer called MIPS presented in the text "Computer Organization and Design" by D. A. Patterson and J. L. Hennessy.