ECE 3401 Lecture 22

Instruction Set Architecture (II)

Overview

- Computer architecture
- Operand addressing
  - Addressing architecture
  - Addressing modes
- Elementary instructions
  - Data transfer instructions
  - Data manipulation instructions
    - Floating point computations
  - Program control instructions
    - Program interrupt and exceptions

Basic Addition Algorithm

- **Steps for addition (or subtraction):**

(1) compute Ye - Xe *(getting ready to align binary point). Ye>Xe*
(2) right shift Xm that many positions to form Xm • 2^{Xe-Ye}
(3) compute Xm • 2^{Xe-Ye} + Ym

Example: .5372400 X 10^2 ⇒ **.5372400 X 10^2**

$^-$ .1580000 X 10^{-1} **– .0001580 X 10^2**

.5370820 X 10^2

if result demands normalization, then normalization step follows:
(4) left shift result, decrement result exponent (e.g., 0.001xx…) right shift result, increment result exponent (e.g., 101.1xx…) continue until MSB of data is 1 (NOTE: Hidden bit in IEEE Standard)
(5) if result is 0 mantissa, may need to zero exponent by special step

Example

- Adding operation on two IEEE single precision floating point numbers (X and Y)

X = 0100 0000 1010 0000 0000 0000 0000 0000
Y = 1100 0000 0011 0000 0000 0000 0000 0000

1	8	23
S	E	M

$N = (-1)^S 2^{E-127} (1.M)$

$X = (-1)^0 2^{129-127} (1.01) = 2^2 * 1.01$
$Y = (-1)^1 2^{128-127} (1.011) = -2 * 1.011$
Xe>Ye
$Y = -2^2 * (1.011 * 2^{-1}) = -2^2 * (0.1011)$
$X+Y = 2^2 * (1.01 - 0.1011) = 2^2 * (0.1001) = 2 * 1.001$
= 0100 0000 0001 0000 0000 0000 0000 0000

Program Control Instructions

- Control over the flow of program execution and a capability of branching to different program segments
- One-address instruction:
  - Jump: direct addressing
  - Branch: relative addressing

Name	Mnemonic
Branch	BR
Jump	JMP
Skip next instruction	SKP
Call Procedure	CALL
Return from procedure	RET
Compare (by subtraction)	CMP
Test (by ANDing)	TEST

Conditional Branching Instructions

- May or may not cause a transfer of control, depending on the value of stored bits in the *PSR (processor state register)*

Branch Condition	Mnemonics	Test condition
Branch if zero	BZ	Z=1
Branch if not zero	BNZ	Z=0
Branch if carry	BC	C=1
Branch if not carry	BNC	C=0
Branch if minus	BN	N=1
Branch if plus	BNN	N=0
Branch if overflow	BV	V=1
Branch if no overflow	BNV	V=0

Conditional Branching Instructions (Contd.)

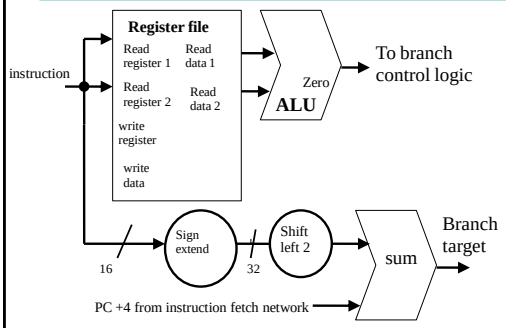
- Unsigned or signed numbers

Branch Condition	Mnemonics	Condition	Status bits
Branch if higher	BH	A>B	C + Z = 0
Branch if higher or equal	BHE	A≥B	C=0
Branch if lower	BL	A<B	C=1
Branch if lower or equal	BLE	A≤B	C + Z =1
Branch if equal	BE	A=B	Z=1
Branch if not equal	BNE	A≠B	Z=0

Branch Condition	Mnemonics	Condition	Status bits
Branch if greater	BG	A>B	(N⊖V) + Z = 0
Branch if greater or equal	BGE	A≥B	N⊖V=0
Branch if less	BL	A<B	N⊖V=1
Branch if less or equal	BLE	A≤B	(N⊖V) + Z =1

8

Datapath for Branch instruction



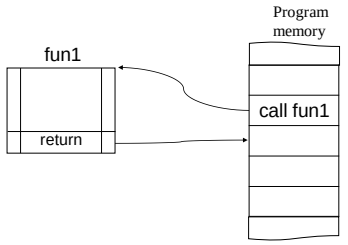
9

Procedure Call and Return Instructions

- Procedure: self-contained sequence of instructions that performs a given computational task
- Call procedure instruction: one-address field
  - Stores the value of the PC (return address) in a temporary location
  - The address in the call procedure instruction is loaded into the PC
- Final instruction in every procedure: return instruction
  - Take the return address and load into the PC
- Temporary Location: fixed memory location, processor register or memory stack
  - E.g. stack
    - Procedure call: SP ← SP-1; M[SP] ← PC+4; PC ← Effective address
    - Return: PC ←M[SP]; SP ←SP+1

10

Procedure Calls



11

Program Interrupt

- Handle a variety of situations that require a departure from the normal program sequence to another service program, similar to a call procedure
- Different from procedure calls:
  - Initiated at an unpredictable point in the program, rather than the execution of an instruction
  - Address of the interrupt service is determined by a hardware procedure
  - The information that defines all or part of the contents of the register set, rather than only the PC, should be stored temporarily
- After finishing interruption, resume to the same state before the interruption
  - *PSR:* other than condition codes, also contains what interrupts allowed, user/system mode indication, etc.

12

Type of Interrupts

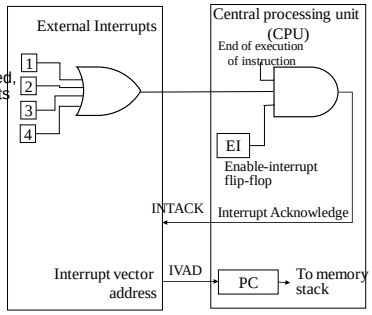
- Hardware interrupts
  - External interrupts:
    - input/output devices requesting transfer of data
    - timing devices time-out event
    - circuit monitoring the power supply detect an impending power failure, in the ISP transfers the register set contents to nondestructive storage like disk, etc.
    - any other external source
  - Internal interrupts (traps):
    - Invalid or erroneous use of an instruction
    - Arithmetic overflow, attempt to divide by zero, an invalid opcode, memory stack overflow, protection violation
- Software interrupts: initiated by executing an instruction
  - System call instructions, change from user mode to system mode

13

Processing External Interrupts

If EI=1, and current instruction is completed, acknowledge interrupts by:

SP ←SP-1
M[SP] ← PC
SP ←SP-1
M[SP] ←PSR
EI←0
INTACK ←1
PC ←IVAD



14