

Microcontrollori

Lezione 2

Leaf procedure(?)

ISA: l'insieme di regole che gestiscono l'interazione tra Hardware e Software “Smaller is Faster” Abbiamo 2 formati, il formato 'R' ed il formato 'L' (Figure 1)

Formato R(32 bit): Si usano 5 bit perché servono ad indirizzare 32 diversi registri.
(OPCODE(6 bit)-rs-rt-rd-sh(shift)-FUNCT)

Formato I: OPCODE-rs-rd-const

MIPS Code:

leaf_example:

1. addi \$sp, \$sp, -4
2. sw \$s0, 0(\$sp) (Save \$s0 on stack)

3. add \$t0, \$a0, \$a1
4. add \$t1, \$a2, \$a3
5. sub \$s0, \$t0, \$t1 (Procedure body)

6. add \$v0, \$s0, \$zero (Result)

7. lw \$s0, 0(\$sp)
8. addi \$sp, \$sp, 4 (Restore \$s0)

9. jr \$ra (Return)

Ogni microprocessore ha un Program Counter che viene aumentato di 4 ad ogni ciclo. Nel caso ci sia un jump. Nel caso di un jmp, il PC sale alla prima istruzione di una possibile funzione. La prima istruzione (in asm) della prima istruzione sarà sicuramente un' "addi" poiché il PC viene 'forzato' ad assumere il valore della posizione di quella funzione.

Local data allocated by callee

Procedure frame (?)

Memory Layout

- Text: program code
- Static data: global variable (static var in C, const arrays and strings)
\$gp initialized to address allowing \pm offsets into this segments
- Dynamic data: heap (malloc in C, new in Java)
- Stack: automatic storage

Jump Addressing

Jump (j and jal) targets could be anywhere in text segment.

Encode full address in instruction

OP-address (6 bits-26 bits) (Fig.3)

Slide: Addressing Mode Summary

Chap. 4 The processor

Exe (execution time) = CCT (clock cycle time) \cdot IC \cdot CPI(= 1)

Control is combinatory Sequenziale è qualcosa che ha uno stato e ne deve tener traccia

Combinatorio è qualcosa il cui output dipende unicamente dagli input

RegWrite: Quando è alto permette di scrivere, quando è basso non scrive

Nelle connessioni c'è sempre voltaggio, non esiste un modo per disconnettere il collegamento, ecco perché serve un controllore per verificare che un certo collegamento sia in uso.

Due numeri sono uguali se la loro sottrazione vale 0, motivo per cui beq(branch if equal) si avvale dell'uso della ALU per l'esecuzione dell'istruzione.

L'istruzione(32 bit) nel datapath viene spaccettata ed inviata.

Per la ALU si fa un controllo "gerarchico", il valore si presenta come una macroclasse di operazioni, ed i 6 bit meno significativi vanno nel controllo della ALU.

Il metodo della pipeline permette di ridurre CCT e CPI contemporaneamente. Mentre una fa l'Instruction fetch, l'altra fa l'accesso ai registri, mentre una accede ai registri l'altra utilizza l'ALU. E' una forma molto low-level di parallelismo.