Assembly CheatSheet

Operazioni sui dati

Nome	Istruzione	**Descrizione**	**Utilizzo**
Load Immediate	`li`	Carica un valore immediato in un registro	`li t0, 10` (Carica 10 nel registro `t0`)
Load Address	`la`	Carica l'indirizzo di un'etichetta in un registro	`la t1, etichetta` (Carica l'indirizzo di `etichetta` in `t1`)
Load Word	`lw`	Carica una parola (32 bit) dalla memoria	`lw t3, 0(t1)` (Carica il valore all'indirizzo contenuto in `t1` nel registro `t3`)
Store Word	`sw`	Salva una parola (32 bit) in memoria	`sw t4, 4(t1)` (Salva il valore di `t4` all'indirizzo `t1 + 4`)
Move	`mv`	Copia il valore di un registro in un altro	`mv t1, t0` (t1 ← t0)
Add Immediate	`addi`	Aggiunge un valore immediato a un registro	`addi t2, t0, 5` (`t2 = t0 + 5`)
Add	`add`	Somma due registri	`add t5, t0, t2` (`t5 = t0 + t2`)
Subtract	`sub`	Sottrae due registri	`sub t6, t0, t2` (`t6 = t0 - t2`)
Multiply	`mul`	Moltiplica due registri, restituendo i 32 bit meno significativi	`mul t6, t0, t1` (`t6 = t0 * t1` a condizione che il risultato rientri in 32 bit)
Load Upper Immediate	`lui`	Carica un valore immediato nei bit più alti del registro	`lui t0, 0x12345` (carica il valore `0x12345000` nei 32 bit più alti del registro `t0`. Questa istruzione è utile per caricare valori grandi che non possono essere espressi

Nome	Istruzione	Descrizione	Utilizzo
			direttamente con un'istruzione immediata, consentendo di costruire valori a 32 bit in due fasi.)
Add Upper Immediate to PC	`auipc`	Carica un valore immediato nei bit più alti del registro sommato al PC	`auipc t0, 0x1` (`t0 = PC + 0x1000`)

Operazioni logiche

Nome	Istruzione	Descrizione	Utilizzo
Shift Left Logical	`sll`	Esegue uno shift logico a sinistra	`sll t7, t3, t4` (`t7 = t3 << t4`)
Shift Right Logical	`srl`	Esegue uno shift logico a destra	`srl t7, t3, t4` (`t7 = t3 >> t4`)
AND	`and`	Effettua un'operazione AND bit a bit	`and t0, t2, t3` (`t0 = t2 & t3`)
OR	`or`	Effettua un'operazione OR bit a bit	`or t0, t2, t3` (`t0 = t2 | t3`)
XOR	`xor`	Effettua un'operazione XOR bit a bit	`xor t0, t2, t3` (`t0 = t2 ^ t3`)

Controllo di flusso

Salto non condizionato: l'esecuzione del programma salta alla riga con l'etichetta corrispondente (`j`, `jal`).

Salto condizionato: l'esecuzione del programma salta alla riga con l'etichetta corrispondente solo se la condizione del salto viene soddisfatta, altrimenti continua normalmente.

Nome	Istruzione	Descrizione	Utilizzo
Jump	`j`	Salta a un'etichetta	`j etichetta`

Nome	Istruzione	Descrizione	Utilizzo
Jump and Link	`jal`	Salta a un'etichetta e salva l'indirizzo di ritorno	`jal ra, funzione` (Salta a `funzione`, salva il ritorno in `ra`)
Branch if Equal to Zero	`bez`	Salta a un'etichetta se il registro è uguale a zero	`bez t0, etichetta` (Salta a `etichetta` se `t0 == 0`)
Branch if Not Equal to Zero	`bnez`	Salta a un'etichetta se il registro è diverso da zero	`bnez t0, etichetta` (Salta a `etichetta` se `t0 != 0`)
Branch if Equal	`beq`	Salta a un'etichetta se due registri sono uguali	`beq t0, t2, etichetta` (Salta a `etichetta` se `t0 == t2`)
Branch if Not Equal	`bne`	Salta a un'etichetta se due registri sono diversi	`bne t0, t2, etichetta` (Salta a `etichetta` se `t0 != t2`)