

Calcolatori Elettronici

Esercitazione 3

M. Sonza Reorda – M. Monetti

M. Rebaudengo – R. Ferrero

L. Sterpone – E. Vacca

Politecnico di Torino

Dipartimento di Automatica e Informatica

Obiettivi

- Operazioni di moltiplicazione
 - Verifica dell'overflow con segno/senza segno
- Lettura di interi da tastiera
 - con segno/senza segno

Istruzioni di moltiplicazione

- Moltiplicazione su 32 bit
`mul rd, rs1, rs2`
- Moltiplicazione unsigned su 64 bit (parte alta)
`mulhu rd, rs1, rs2`
- Moltiplicazione signed su 64 bit (parte alta)
`mulh rd, rs1, rs2`
- Moltiplicazione su 64 bit tra signed e unsigned (parte alta)
`mulhsu rd, rs1, rs2`

Esercizio 1

- Siano inizializzate in memoria le variabili word:
 - `op1 : .word 0x0000D000`
 - `op2 : .word 0xFFFFD000`
- Considerando le due variabili come unsigned, per ognuna delle seguenti moltiplicazioni si visualizzi il risultato oppure si segnali l'overflow
 - `op1 * op1`
 - `op2 * op2`
 - `op1 * op2`

Esercizio 1: risultato

- L'output atteso è:

2835349504

overflow

overflow

Esercizio 2

- Siano inizializzate in memoria le variabili word:
 - `op1: .word 0x0000D000`
 - `op2: .word 0xFFFFD000`
- Considerando le due variabili come signed, per ognuna delle seguenti moltiplicazioni si visualizzi il risultato oppure si segnali l'overflow
 - `op1 * op1`
 - `op2 * op2`
 - `op1 * op2`

Esercizio 2: risultato

- L'output atteso è:

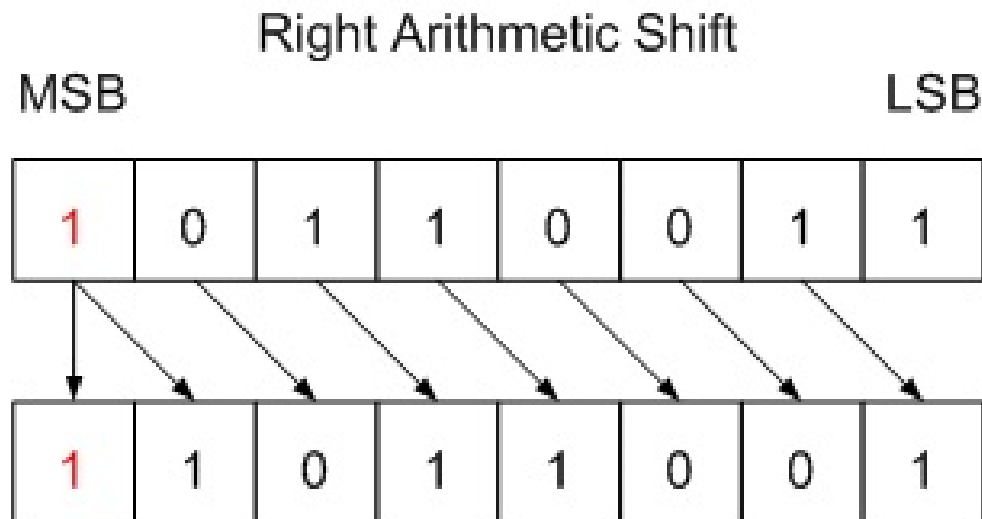
`overflow`

`150994944`

`-654311424`

SRA - Shift Right Arithmetic

Arithmetic: the sign bit shifts into the most significant bit and the other bits (including the sign bit) shift of one position to the right



Overflow della moltiplicazione

- La moltiplicazione tra numeri su 32 bit dà un risultato fino a 64 bit
- La moltiplicazione fra numeri in binario puro (unsigned) è rappresentabile su 32 bit se i 32 bit alti (più significativi) sono zero
- La moltiplicazione fra numeri in complemento a 2 (signed) è rappresentabile su 32 bit se:
 - i 33 bit alti sono 0 (il risultato è positivo)
 - i 33 bit alti sono 1 (il risultato è negativo)

Esercizio 3

- La ecall 63 legge in input una stringa
- Si realizzi un programma che determini se la stringa letta corrisponda ad un numero valido (ossia se tutti i caratteri siano cifre)

Esercizio 4

- Si modifichi l'esercizio 3 per la lettura di un numero intero unsigned tramite la ecall 63.
- Oltre a verificare se i caratteri introdotti siano cifre, il programma deve controllare se il numero sia rappresentabile su 4 byte in binario puro.
- Il numero introdotto in input deve essere memorizzato in una variabile word (se possibile) oppure un opportuno messaggio deve indicare che il numero è troppo grande.

Implementazione

- Per convertire una sequenza di caratteri in un intero si utilizza un ciclo. Dopo aver inizialmente azzerato un registro accumulatore, ad ogni iterazione:
 1. l'ultimo carattere letto è convertito in intero sottraendo al suo codice ASCII il valore '0'
 2. il valore nell'accumulatore è moltiplicato per 10
 3. si somma il valore calcolato all'accumulatore.
- Si noti che le operazioni ai punti 2 e 3 possono dare un overflow. In questo caso il programma deve stampare un opportuno messaggio.

Esempio

- L'utente inserisce i caratteri '3', '4', '6', '\n'
- Prima iterazione:
 1. valore letto = '3' - '0' = 3
 2. accumulatore * 10 = 0 * 10 = 0
 3. valore corrente = 0 + 3 = 3
- Seconda iterazione:
 1. valore letto = '4' - '0' = 4
 2. accumulatore * 10 = 3 * 10 = 30
 3. valore corrente = 30 + 4 = 34
- Terza iterazione:
 1. valore letto = '6' - '0' = 6
 2. valore precedente * 10 = 34 * 10 = 340
 3. valore corrente = 340 + 6 = 346
- Quarta iterazione:
 1. valore letto = '\n'
 2. Il programma termina e stampa a video 346

Esercizio 5

- Si modifichi l'esercizio 4 per la lettura di un numero intero signed tramite la ecall 63.
- Il programma deve controllare che:
 - il primo carattere introdotto dall'utente sia '+' o '-'
 - tutti i caratteri successivi siano numeri
 - il numero sia rappresentabile su 4 byte in Ca2
- Il numero introdotto in input deve essere memorizzato in una variabile word (se possibile) oppure un opportuno messaggio deve indicare che il numero è troppo grande.
- Esempio di input: "-346" seguito da invio