Architettura degli Elaboratori

December 4, 2018

1 Introduzione

Ci tengo a ricordare che questi sono solo esempi e non rappresentano in alcun modo l'esatto contenuto dell'esame in questione.

Ogni esempio contenuto in questo documento e' stato realizzato da Mirco De Marchi.

2 Codifica dell'informazione

2.1 Meno Infinito

Dire come viene rappresentata la grandezza meno infinito in virgola mobile secondo lo standard IEEE 754 in singola precisione. Quali sono i motivi alla base della scelta di questa rappresentazione?

Rappresentazione di meno infinito in virgola mobile.

Il primo bit, il segno, riporta 1, cioè segno negativo, tutti i bit dell'esponente sono 1 e tutti i bit della mantissa sono 0.

Convertendo il numero ottengo:

segno: negativo

esponente: 255 in modulo, che convertito in eccesso 127 diventa 255-127= +128 mantissa: 1.0₁₀ perché la parte parte frazionaria non riporta nessun numero. Quindi il numero ottenuto è -1.0 * 2⁺¹²⁸, che in virgola mobile singola precisione è in assoluto il numero più piccolo rappresentabile, che dunque può solo che essere -infinito. Infatti in realtà il numero più piccolo esprimibile con la codifica virgola mobile singola precisione è formato da tutti 1, tranne per uno 0 nel bit meno significativo dell'esponente e corrisponde in base 10 a un valore che si avvicina molto a -2 * 2⁺¹²⁷, che è uguale a -2⁺¹²⁸.

2.2 Somma in complemento a 2

Rappresentare e sommare in complemento a due su 6 bit i due numeri relativi +24 e -13.

- +24 è positivo, dunque è sufficiente convertirlo in modulo per averlo in complemento a due e corrisponde a 011000₂.
- -13 invece è negativo, quindi prima lo converto come se fosse +13, cioè 001101₂, poi faccio il negativo, che è 110010, sommo 1 e ottengo 110011₂, che è proprio -13 in complemento a due.

Eseguo la somma: $011000 + 110011 = 001011_2$, che corrisponde a $+11_{10}$.

Questa volta non è necessario fare l'estensione di segno poiché sto lavorando con tutti i numeri a 6 bit.

Facendo la verifica della somma: 24 + (-13) = +11₁₀

2.3 Cambio da Complemento a 2 a Modulo

Quale sarebbe stato il risultato delle somma interpretando i codici ottenuti come numeri in modulo? E quale interpretandoli in modulo e segno?

011000₂, corrisponde a +24 sia in modulo, sia in modulo+segno. 110011₂, corrisponde come riportato sopra a -13 in complemento a due, in modulo corrisponde a (1+2+16+32)=51, in modulo+segno corrisponde a -(1+2+16)=-19. quindi mi aspetto che la somma in modulo sia uguale a 24+51=75 e che la somma in modulo+segno sia uguale a 24+(-19)=+5 Procedendo con la somma:

in modulo: 011000 + 110011 = 1001011 che convertito in base 10 effettivamente corrisponde a +75₁₀, tuttavia in realtà il vero risultato è questo: 001011, perché sto lavorando con 6 bit. Questo è un caso di overflow poiché il numero da rappresentare necessita un bit in più.

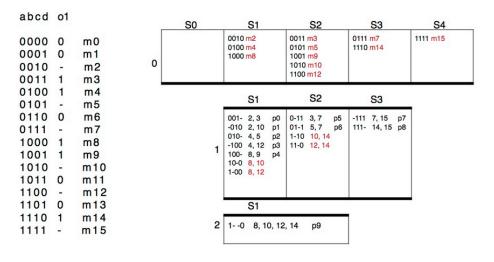
In modulo+segno, effettuando la somma normalmente avviene questo: 011000 + 110011 = 001011, che corrisponde a $+(1+2+8)=+11_{10}$. Il risultato è chiaramente sbagliato. Questo perché la somma in modulo+segno è più complicata e va effettuata in questo modo: se i segni sono discordi (che è il nostro caso) bisogna controllare in valore assoluto qual è il valore più alto: in questo caso il valore più alto è +24, che essendo positivo dà come risultato un valore positivo. Dopodiché bisogna procedere con la sottrazione dei due codici in valore assoluto e sottrarre il minore dal maggiore: 11000 - 10011 = 001012, con il segno (positivo) diventa: 0001012, che corrisponde proprio a +510.

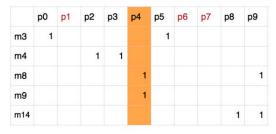
3 Macchine a Stati Finiti

Esempio fornito nel PDF: FSM.pdf

4 Semplificazione Quine-McCluskey

Calcolare con il metodo di Quine McCluskey gli implicanti primi essenziali della seguente funzione f(a, b, c, d): ON-SET = $\{m3, m4, m8, m9, m14\}$, DC-SET = $\{m2, m5, m7, m12, m10, m15\}$.





Implicante primo essenziale: p4 = m8, m9

a*!b*!c