Algoritmi Numerici (Parte I)

[Lezione 6] Operazioni aritmetiche

Alessandro Antonucci
alessandro.antonucci@supsi.ch

https://colab.research.google.com/drive/10HxMjR7A0Qa8X3cRe2ZCgO-rtRyYvVKY

Somma

- Sistema a 4 bit con complemento a 2
- Somma con circuiti con porte Booleane (link)
  (es. $\mathbf{0}+\mathbf{0}=\mathbf{1}+\mathbf{1}=\mathbf{0}$ e $\mathbf{0}+\mathbf{1}=\mathbf{1}+\mathbf{0}=\mathbf{1}$ è XOR)
- $\mathbf{0010}+\mathbf{0011}=\mathbf{0101}$ $2+3=5$
- $\mathbf{0010}+\mathbf{1100}=\mathbf{1110}$ $2+(-4)=-2$
- $\mathbf{0011}+\mathbf{0110}=\mathbf{1001}$ $3+6\neq-7$ overflow!
- $\mathbf{1100}+\mathbf{1010}=\mathbf{0110}$ $-4-6\neq10$ underflow!

Over/under flow se il primo bit è uguale

nei due addendi, ma diverso nella somma

- $\mathbf{1100}+\mathbf{1100}=\mathbf{1000}$ $-4-4=-8$
  (giusto anche con riporto)

Sottrazione (e precisione multipla)

- Somma di due numeri a 4 bit:
  - Due addendi (4+4 bit)
  - Carry (C, 1 bit) si attiva per il riporto $\mathbf{1} + \mathbf{1} = \mathbf{1}$ è AND
  - Overflow (V, 1 bit) si attiva per identificare over/under
- Numeri piu grandi
  - Precisione multipla (tante locazioni affiancate)
  - Si inizializza $\boldsymbol{C} = \mathbf{0}$ (non importa se alla fine $\boldsymbol{C} = \mathbf{1}$)
  - Risultato corretto solo se $\boldsymbol{V} = \mathbf{0}$
- Sottrazione
  - a - b = a + (-b)
  - Per scrivere $-\boldsymbol{b}$, scrivo $\boldsymbol{b}$ in base 2 e poi cambio segno (ricopio da dx finché non trovo $\mathbf{1}$, scrivo quello, poi nego)

Moltiplicazione e Divisione

- Moltiplicazioni in colonna = somme numeri spostati a sx
- ASL (arithmetic shift left) sposto i bit a sx (uno zero a dx)
- Es. ASR(0011)=0110
- È moltiplicazione per 2 (se $V = 0$ vale anche con complemento a 2)
- Divisione: analoga ma sposto a dx
- ASR (arithmetic shift right), divisione (intera) per 2

Aritmetica float

- Dati due numeri (float)
  - $a = m_1 \cdot b^q$
  - $b = m_2 \cdot b^p$ $(q > p)$
- Posso calcolare
  - $a + b = (m_1 + m_2 \cdot b^{p-q}) \cdot b^q$
  - $a \cdot b = (m_1 \cdot m_2) \cdot b^{q+p}$
  - $a/b = (m_1/m_2) \cdot b^{q-p}$
- Le operazioni sulla mantissa introducono un errore (riducono le cifre significative)

Under/over flow con numeri float

Formati float-like caratterizzati da:

- m_1 numero più piccolo rappresentabile
- M_1 numero più grande.

Quando inserisco un numero x, il sistema restituisce

- overflow se x esterno a $[-M_1, M_1]$
- underflow se interno a $(-m_1, m_1)$ (approssimato con lo zero)

Nel formato float, se tutti i bit dell'esponente sono uno, casi speciali

Es. $0|11111111|00000000000000000000000 \rightarrow +\infty$ Es. $0|11111111|00000000010000010000000 \rightarrow NaN$