Note Title 23-08-2012

Aug 23rd - Ropar

ARM → Machine Insts. [ARCHITECTURE]

Basic Elements: [ORGANIZATION]

ALU: adder, shifter, multiplier divider.

Memory: SRAM, DRAM, Flip-Flops

Adder

1 bit addition.

[HALF ADDER]

$$\begin{array}{r} c_1 \quad \\ a_1 \\ + \; b_1 \\ \hline S_1 \end{array}$$

$\oplus \rightarrow$ exclusive or

$S_1 = a_1 \oplus b_1$

$\cdot \rightarrow$ and

$C_1 = a_1 \cdot b_1$

Full Adder.

$$\begin{array}{rr} c_1 & \overset{c_0}{a_1} \\ + & b_1 \\ \hline & s_1 \end{array}$$

$+ \rightarrow OR$

$$s_1 = a_1 \oplus b_1 \oplus c_0$$

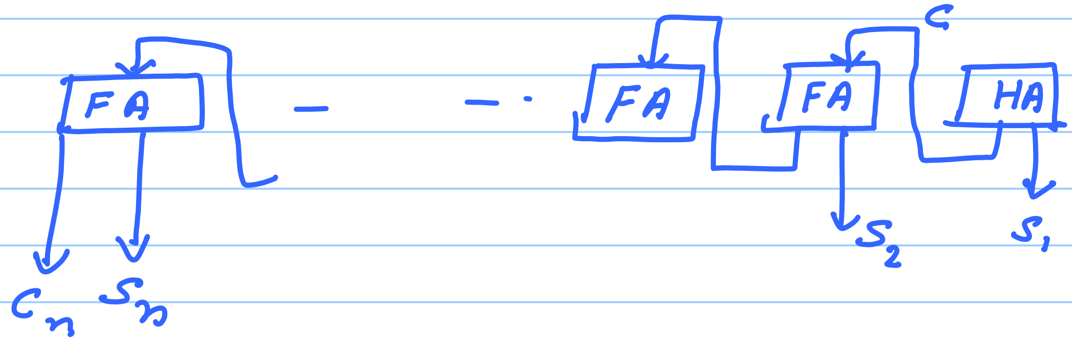
$$c_1 = a_1 . b_1 + a_1 c_0 + b_1 c_0$$

n-bit addition

$$\begin{array}{rrr} & a_n & \cdots\cdots & a_1 \\ + & b_n & \cdots\cdots & b_1 \\ \hline & & & \\ \end{array}$$

$$\underbrace{c_n \quad s_n \quad \cdots\cdots \quad s_1}_{n+1}$$

Ripple Carry Adder.



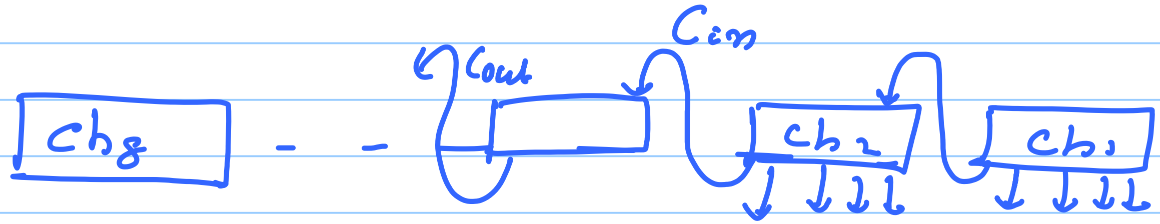
Carry is rippling from the first to last adder

Time it takes to do addition : $O(n)$
[slow]

Reason for slow operation: The carry is propagating from position $1 \ldots n$

Carry - Select Adder

$$\begin{array}{cccccc} & \text{chunk 8} & \text{chunk 7} & & & \text{chunk 1} \\ & \left[a_{32} \cdots a_{29} \right] & \left[a_{28} \cdots \right] & - & - & \left[a_4\ a_3\ a_2\ a_1 \right] \\ + & \left[b_{32} \cdots b_{29} \right] & \left[b_{28} \cdots \right] & - & - & \left[b_4\ b_3\ b_2\ b_1 \right] \\ \hline c_{32} & s_{32} \cdots & & & & s_1 \end{array}$$



Carry Select Adder.

Stage 1:

$\forall$ chunks,

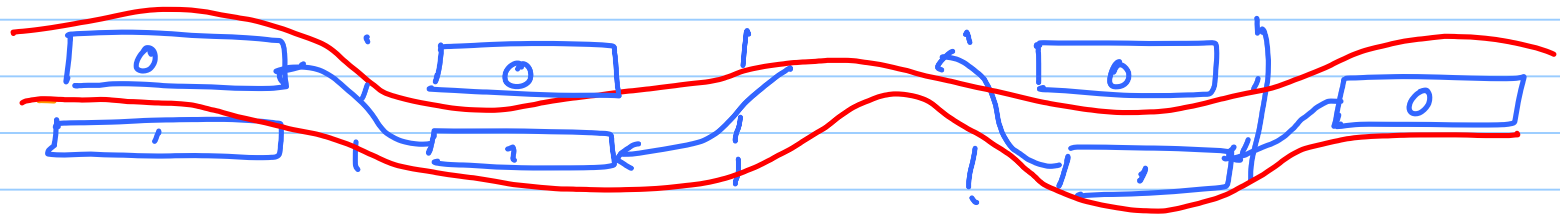
compute two results

$result_1$ (assuming $C_{in} = 1$)
(sum bits & C_{out})
$result_0$ (assuming $C_{in} = 0$)

Chunk size is k

Time ($Stage_1$) : $O(k)$

Stage 2 :
Make the carry ripple across chunks



Time ($Stage_2$) = #chunks : n/k

Time: $\left\{k + \frac{n}{k}\right\}$

E.g. 8 bit addition chunk size is 2

$$\begin{array}{cccc} C_4 & C_3 & C_2 & C_1 \\ 10 & 11 & 00 & 01 \\ 11 & 01 & 00 & 11 \\ \hline 1\,10 & 00 & 01 & 00 \end{array}$$

Stage 1

C_4	C_3	C_2	C_1
101	100	000	100
110	101	001	

C_1: 1 = C, 00 = Sum

Stage 2

$C_{in} = 0$	101	100	000	100
$C_{in} = 1$	110	101	001	

$1 \quad \overline{10 \quad 00 \quad 01 \quad 00}$

Time: $k + n/k$

optimal chunk size: $\sqrt{n}$

$$f(k) = k + \frac{n}{k}$$

$$\frac{\partial f(k)}{\partial k} = 1 - \frac{n}{k^2} = 0$$

$$\Rightarrow k = \sqrt{n}$$

Time: $2\sqrt{n} \leftrightarrow O(\sqrt{n})$

<u>Extension</u>

Time waste:

A unit (chunk) is idle in the

2nd stage after it has done its job.

variable sized chunk.

$\boxed{m} \quad \cdots \quad \boxed{i+1} \quad \boxed{i} \; \cdot \quad \boxed{2} \quad \boxed{1}$

$$\sum_{1}^{m} i = n$$

Operation:

1) Every chunk computes two results ($C_{in} = 0$, $C_{in} = 1$)

2) Claim: The moment it is done with its computation it gets the right value of C_{in}

<u>Proof</u>

Mathematical Induction

Base Case: $chunk_1$

$C_{in} = 0$ ✓

Induction Case:

i^{th} Chunk

At time (i) → finishes its computation.

By induction hypothesis, the previous chunk finishes its computation in time $(i-1)$ and forwards the correct C_{in} to chunk i by time i.

→ Before time $(i+1)$, compute the right value of C_{in} for $chunk_{(i+1)}$ □

Total time: $O(m)$

$$\sum_{1}^{m} i = n$$

$$\Rightarrow \frac{m(m+1)}{2} = n \qquad m = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\Rightarrow m^2 + m - 2n = 0$$

$$= \frac{-1 \pm \sqrt{1 + 8n}}{2}$$

$$m = \frac{\sqrt{8n+1} - 1}{2}$$

$$= \sqrt{2n + 1/4} - 1/2 \approx \sqrt{2} \cdot \sqrt{n}$$

Time: $O(m) = O(\sqrt{n})$ (complexity is same as fixed chunk algo.)

Time :

Algorithm	Exact Time	Asymptotic Time
1) Ripple Carry Adder	$\approx n$	$O(n)$
2) Carry Select Adder (Fixed size chunk)	$\approx 2\sqrt{n}$	$O(\sqrt{n})$
3) Carry Select Adder (Variable size chunk)	$\approx \sqrt{2} \cdot \sqrt{n}$	$O(\sqrt{n})$
4) Carry Lookahead Adder	$\approx 2 \cdot \log(n)$	$O(\log(n))$

Carry Lookahead Adder

$0 \leftarrow C_{in}$

$+\ 0$

0

$C_{out} = 0$

[absorb]

$0 \leftarrow C_{in}$

$+\ 1$

$C_{out}\quad S$

$C_{out} = C_{in}$

[propagate]

C_{in}

1

$+\ 1$

$1\ S$

C_{out}

$C_{out} = 1$

[generate]

a_1

$+\ b_1$

$\cdot\ \cdot\ \cdot$

$G_1 = a_1 \cdot b_1$ (generate)

$P_1 = a_1 \oplus b_1$ (propagate)

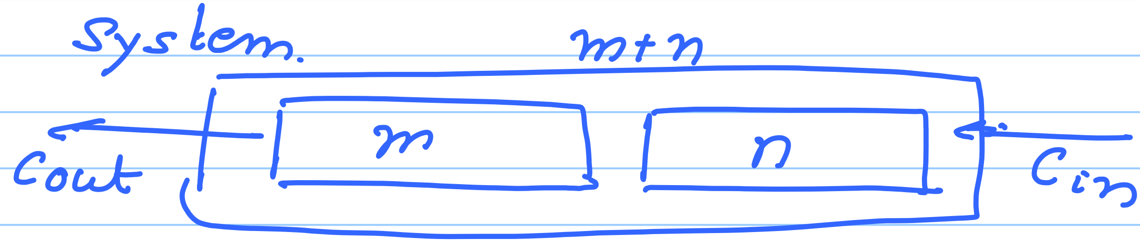
$$
\begin{array}{rcc}
 & a_2 & a_1 \\
+ & b_2 & b_1 \\
\hline
\_ & \_ & \_
\end{array}
$$

$C_{out} \leftarrow \boxed{\text{chunk}} \leftarrow C_{in}$

$$C_{out} = G + P \cdot C_{in}$$

For a 1-bit chunk:
we have equations for G and P

multibit system.



$$
\left\|\begin{array}{l}
G_{m+n} = G_m + P_m \cdot G_n \\
P_{m+n} = P_m \cdot P_n
\end{array}\right. \quad - - - - \quad (1)
$$

E.g.

$$\begin{bmatrix} a_2 & a_1 \\ b_2 & b_1 \end{bmatrix}$$

$$\begin{array}{rrr} - & - & - \\ & 1 & 0 \\ + & 1 & 1 \\ \hline 1 & 1 & 1 \end{array}$$

$$\begin{array}{rrr} & 1 & 1 \\ & 0 & 1 \\ \hline 1 & 0 & 0 \end{array}$$

$$G = a_2 . b_2 + (a_2 \oplus b_2) . a_1 . b_1$$

$$P = (a_1 \oplus b_1) . (a_2 \oplus b_2)$$

$$G_1 = a_1 . b_1 \qquad P_1 = a_1 \oplus b_1$$

$$G_2 = a_2 . b_2 \qquad P_2 = a_2 \oplus b_2$$

$$\begin{cases} G = G_2 + P_2 . G_1 & \quad -- \; (2) \\ P = P_2 . P_1 \end{cases}$$

🚩 TODO: Go to the web and understand this

Tutorial This week: Sunday (10 - 1 pm)

Tomorrow

Carry Lookahead Adder.