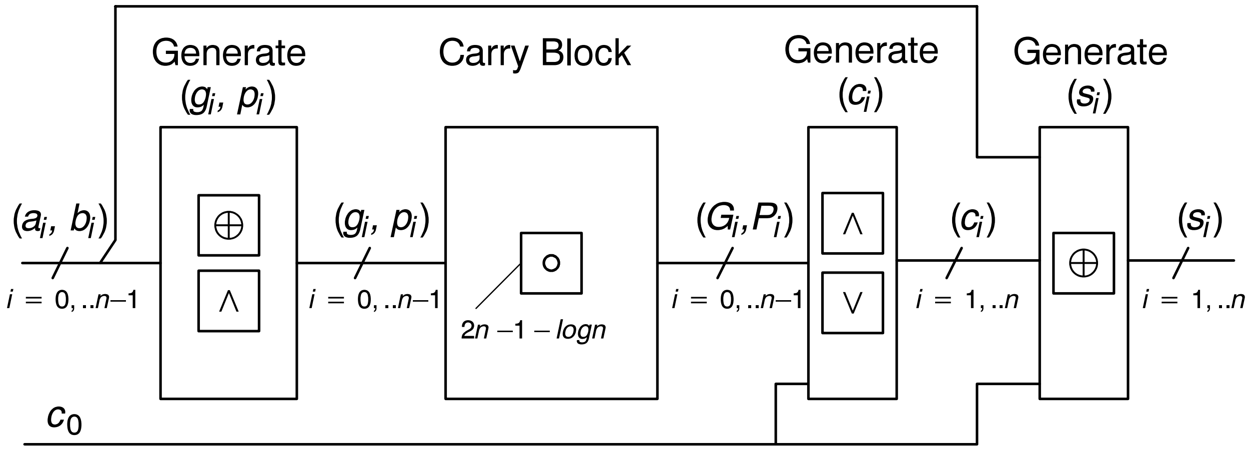
Solutions for Chapter 4

Exercise 4.1

a. Binary carry lookahead adder (BLC adder)



n : word width

Number of cuts: $1 + \log n + \log\frac{n}{2} - 1 = 2\log n - 1$

$a_i, b_i \rightarrow g_i, p_i + stages_{\text{bin.tree}} + stages_{\text{inv.bin.tree}} - 1$ (1 stage overlap)

Number of lines/cut: $2n + 1 + 2n = 4n + 1$

$lines_{a_i, b_i \rightarrow s_i} + lines_{c_0} + lines_{g_i, p_i}$

$$\#FFs = (4n + 1) \cdot (2\log n - 1)$$

Throughput determined by max. delay of the operands:

$$(g_{i,i+1}, p_{i,i+1}) \rightarrow (G_i, P_i)$$

$$T_{D,carryblock}(F = 1) = (7 + 2)\tau = 9\tau$$

$$T_D = 9\tau + T_{D,FF}$$

b. Conditional sum adder (CS adder)

For the word width n with $k = 2^n - 1$: in each stage the number of lines/cut can be determined by the following components:

#stages:	$1, 2, .., \log(n + 1)$
#lines/cut:	$2, 3, 5, 9, ..2^i + 1$
#cuts	$2 \cdot n, .., 2 \cdot 7, 2 \cdot 3, 2 \cdot 1$

$$= 2 \cdot \left(2^{log(n+1)}–i–1\right)$$

$$= 2\left((n+1) \cdot 2^{-i}–1\right)$$

remaining cuts: 1, 2, 4, 8, ..., 2^{i}

$$\#FFs = \sum_{i=0}^{\log(n+1)-1} \left[2 \cdot \left((n+1) \cdot 2^{-i}–1\right) \cdot \left(2^{i}+1\right) + 2^{i}\right]$$

$$= \sum_{i=1}^{\log(n+1)} \left[2 \cdot (n+1) \cdot \left(2^{-(i-1)}+1\right) - 2^{i-1} - 2\right]$$

with: $\sum_{i=1}^{\log(n+1)} 2^{i-1} = n$ and $\sum_{i=1}^{\log(n+1)} 2^{-(i-1)} = \frac{2n}{n+1}$

$$\#FFs = 2n \cdot log(n+1) + 3n$$

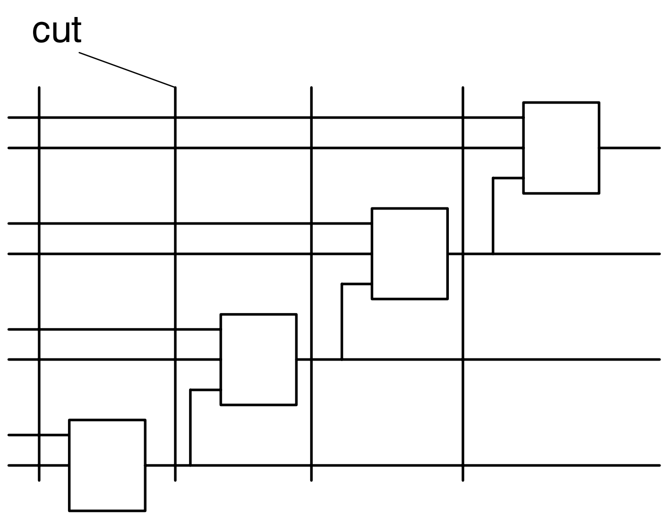
Critical path of the carry block determined by max. of CC–cell and MUX($\frac{n+1}{2} + 1$):

$$T_{D,carryblock}(F = 1) = (10 + F + log(n+1))\tau$$

$$= (11 + log(n+1))\tau$$

$$T_D = (11 + log(n+1))\tau + T_{D,FF}$$

c. Pipelined Ripple carry adder (RCA)



Number of cuts: $\sum_{i=1}^{n-1} n + n–1$

$$a_i, b_i \rightarrow s_i \text{ and } c_i$$

Number of lines/cut: 3: a_i, b_i and s_i and 1 for c_i

$$\#FFs = \frac{3n^2 - n - 2}{2}$$

Critical path of the carry block $a_{i,}b_i \rightarrow s_i \hat{=} XOR + FF$ (according to (3.2.12)):

$$T_{D,FA} = 24\tau$$

$$T_D = 24\tau + T_{D,FF}$$

Comparison:

Considering (3.2.52–54) and the results of the above pipelined adders, one can say, that for large n the pipelined BLC adder is more efficient than the other ones. Whereas for moderate n the RCA is the adder with the highest efficiency.

Exercise 4.2

The word width of the accumulators results to:

$ww = \lceil log((2^m - 1)N + 1) \rceil \approx m + \log n$.

a. Under the assumption of $N_{FA} = 26$ transistors and $N_{FF} = 8$ transistors the following transistor counts result:

CPA: $N_{CPA} = N_{FA}ww + N_{FF}ww = 34 \cdot ww$

The CSA accumulator contains the components :

– CSA (word width: *ww*)

– FFs for the carry and sum vector

– CPA (RCA, word width: *ww*)

CSA: $N_{CSA} = N_{FA}(ww - 1) + 2N_{FF}(ww - 1) + N_{FA}(ww - 1)$

$N_{CSA} = 68(ww - 1)$

$A_{CSA} \approx 2\,A_{CPA}$

b. Total computation time in case of CPA accumulator:

The delay of one computation can be derived from the path:

$a_i, b_i \rightarrow s_i, c_{i+1}$

$(ww - 1) \times c_i \rightarrow s_{i,}c_i$

1 register

$$T_{D,CPA} = T_{D,FA} + (ww-1)\frac{T_{D,FA}}{2} + T_{D,FF}$$
$$= T_{D,FA}\left(\frac{ww}{2} + 1\right)$$
$$T_{D,ACC} = N\,T_{D,CPA}$$

The computation time of the CSA accumulator, consists of N computations in the CSA, the delay in the register as well as one computation for the carry and sum vector in the final CPA.

$$T_{D,CSA,1} = T_{D,FA} + T_{D,FF} = \frac{3}{2}T_{D,FA}$$
$$T_{D,CSA,2} = T_{D,FA} + (ww - 1)\frac{T_{D,FA}}{2} = T_{D,FA}\frac{(ww + 1)}{2}$$
$$T_{D,ACC} = \frac{T_{D,FA}}{2}(3N + ww + 1)$$

c. According to (4.2.5) the efficiency is defined as : $\eta = \frac{1}{TA}$

If the CSA accumulator should be more efficient than the CPA accumulator the following inequation must be fulfilled:

$$\eta_{CSA} > \eta_{CPA}$$
$$N\left(\frac{ww}{2} + 1\right) > ww + 3N + 1$$
$$ww > \frac{4N + 2}{N - 2}$$

To large N's applies: $ww > 4$

or approximately: $m > 4 - \log N$

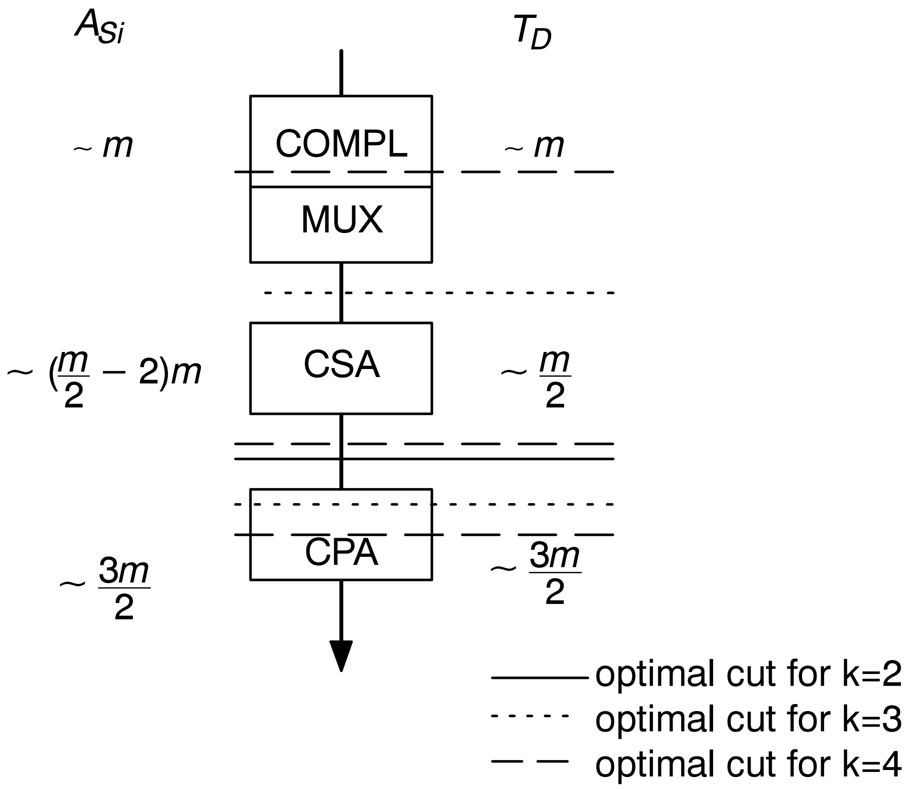
Exercise 4.3

a. The delay of the Booth multiplier is determined to:

$$T_{D,Booth} = T_{D,Compl} + T_{D,CSAtree} + T_{D,CPA}$$
$$T_{D,Compl} = T_{D,FA} \cdot m$$
$$T_{D,CSAtree} = T_{D,FA} \cdot \left(\frac{m}{2} - 2\right) \approx T_{D,FA}\frac{m}{2}$$
$$T_{D,CPA} = T_{D,FA} \cdot (2m - (\frac{m}{2} - 2 + 1)) = T_{D,FA} \cdot (3\frac{m}{2} + 1) \approx T_{D,FA} \cdot \frac{3}{2}m$$

$$T_{D,Booth} \approx T_{D,FA} \cdot 3m$$

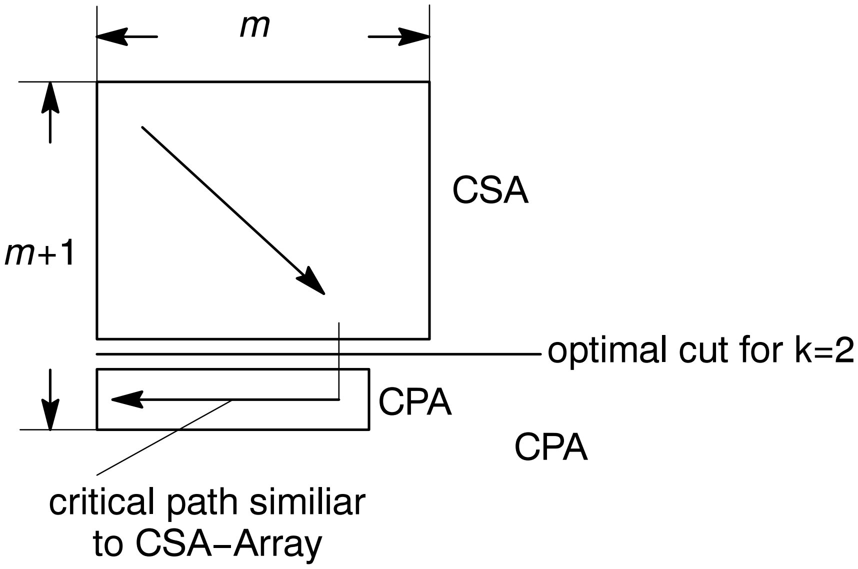
The cut−lines can be determined by dividing $T_{D,Booth}$ by factors of 2, 3 and 4.



The optimal cuts for k are depicted in the figure above. For k=3 its beeing optimal using cut within CPA. If using natural border, CPA determines throughput. For k = 4 optimal cuts within complementer (COMPL) and CPA.

b. According to section 3.3.2, the delay of a Pezaris array multiplier is determined to:

$$T_{D,Pez} \approx 2mT_{D,FA}$$



The optimal cut for k =2 is depicted in the figure above. For k=3 cut within CSA− and CPA−Array. For k = 4 or multiple of 2 cut within CSA and CPA.

c. With respect to k=2,3,4, the silicon areas for the Booth and the Pezaris array multiplier are approximately given by:

$$A_{Booth} \approx \frac{m^2}{2} A_{FA} + kmA_{FF}$$

$$A_{Pez} \approx (m^2 + m) A_{FA} + 2mA_{FF}$$

Then the efficiencies are:

$$\eta_{Booth}(k) = \frac{1}{\left(\frac{3}{k} mT_{D,FA} + T_{D,FF}\right) \cdot \left(\frac{m^2}{2} A_{FA} + kmA_{FF}\right)}$$

$$\eta_{Pez}(k) = \frac{1}{\left(\frac{2}{k} mT_{D,FA} + T_{D,FF}\right) \cdot \left((m^2 + m) A_{FA} + 2mA_{FF}\right)}$$

Due to that, for large m a pipelined Booth array multiplier should be prefered. However, the Pezaris multiplier is more suitable for an automatic insertation of pipeline stages, because of its regular structure.

Exercise 4.4

According to example 4.2.2 and for $\gamma = \frac{1}{N}$ the ξ functions are as follows:

$$\xi_R = \frac{R_N}{R_0} = N \cdot (1.4 \cdot \gamma - 0.4)$$

$$\xi_A = \frac{A_N}{A_0} = N$$

$$\xi_P = \frac{P_N}{P_0} = N \cdot \gamma^2 \cdot (1.4 \cdot \gamma - 0.4)$$

Then the cost or target functions becomes to:

$$q_R = \begin{cases} \frac{1}{3}\xi_R & 0 < \xi_R \leq 3 \\ 1 & 3 < \xi_R \end{cases}$$

$$q_A = \begin{cases} 1 - \frac{1}{6}\xi_A & 0 < \xi_A \leq 6 \\ 0 & 6 < \xi_A \end{cases}$$

$$q_P = \begin{cases} 1 - \frac{2}{3}\xi_P & 0 < \xi_P \leq \frac{3}{2} \\ 0 & \frac{3}{2} < \xi_P \end{cases}$$

With respect to (4.2.34) it can be determined:

$$\zeta = q_R \cdot q_A \cdot q_P$$

$$\zeta = \frac{N}{3} \cdot (1.4\gamma - 0.4) \cdot (1 - \frac{N}{6}) \cdot \left(1 - \frac{2N}{3}\gamma^2(1.4\gamma - 0.4)\right) \qquad N \leq 6$$

$$\zeta = \left(\frac{N}{3} - \frac{2}{9}N^2\gamma^2 + \frac{N^2}{18}\right) \cdot (1.4\gamma - 0.4) + \frac{2N^3}{54}\gamma^2(1.4\gamma - 0.4)^2$$

Partial derivatives:

$$\frac{\delta\zeta}{\delta N} = \left(\frac{1}{3} - \frac{4}{9}N\gamma^2 + \frac{4}{9}N\right) \cdot (1.4\gamma - 0.4) + \frac{1}{9}N^2\gamma^2 \cdot (1.4\gamma - 0.4)^2$$

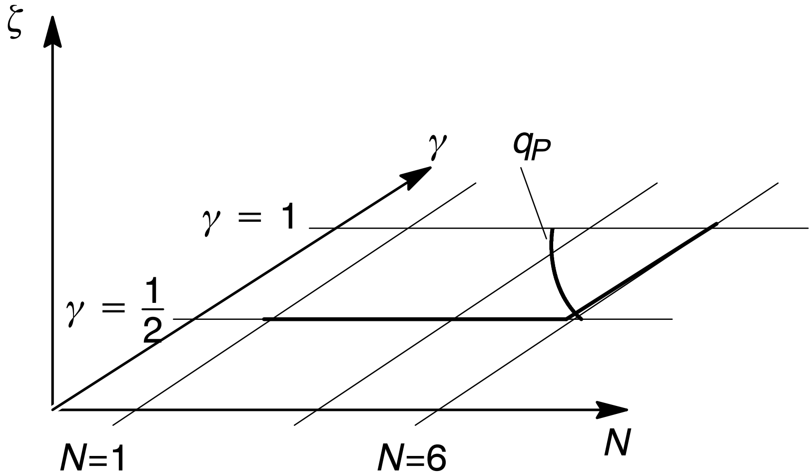
$$\frac{\delta\zeta}{\delta\gamma} = \left(\frac{N}{3} - \frac{2}{9}N^2\gamma^2 + \frac{N^2}{18}\right) \cdot \frac{7}{5}\gamma + \frac{N^3}{27}\gamma^2(1.4\gamma - 0.4)^2$$

Roots of the partial derivatives are in the linear region of all 3 target functions.

$$\textit{boundary } \gamma: \ \gamma = 1 \textit{ or } \gamma = \frac{1}{2} \ \rightarrow \ N = ?$$

$$\textit{boundary } N: \ N = 1 \textit{ or } N = 6$$

Search at boundary lines of defined region.



$$\gamma = 1, \ N = 1: \quad \zeta = \frac{1}{3} \cdot \frac{5}{6} \cdot \frac{1}{3}$$

$$\gamma = 0.5, N = 1: \quad \zeta = \frac{1}{10} \cdot \frac{5}{6} \cdot \frac{3}{40}$$

$$\vdots \qquad \qquad \vdots$$