Lecture ”HW-SW Codesign”

Exercise I

Pipelining

What is the aim of pipelining? Which problems do occur?

With the help of pipelining the throughput of a system can be increased. Figure 1 shows a system which should add the 3 vectors \( \mathbf{a}, \mathbf{b}, \mathbf{c} \) forming the resulting vector \( \mathbf{y} \). Each vector contains \( N = 20 \) samples and the internal delay of one adder is assumed to be \( T_{Add} = 10 \) ns. The maximum clock frequency of the system is determined by the sum of all partial delays, i.e., in our case

\[
T_{total} = \sum T_i = \frac{1}{2T_{Add}} = \frac{1}{f_{clk}} = 50 \text{ MHz}
\]  

(1)

Hence the total time required to compute all 20 values is 400 ns. If we introduce pipeline registers as shown in Fig. 1b) the delay between the input register and the pipeline register and from there to the output register is only \( T_{Add} \). The system could be run at a clock speed of \( f_{clk} = 100 \) MHz and the total computing time reduces to \( T_c = (N + 1)/f_{clk} = 210 \) ns. (The pipeline register causes 1 clock cycle latency).

Note: Pipelining does not decrease the actual computing time of a single value. In our example it still takes 20 ns to compute a value but the rate at which a result is produced was doubled.

Exercises

1. A system is described by the signal flow graph shown in Fig. 2. Each node represents a logical operation, all nodes should have the same delay.
Please introduce pipelining such that the length of the critical path \((c.p.)\) becomes a minimum when
(a) The output latency is the same for all outputs (general case)
(b) What happens when the cut-set lines are allowed vertically only

2. A 4-Bit carry-ripple adder should be realized. Define the c.p. of that circuit. How many pipeline registers are necessary to reduce the c.p. to one full adder cell (FA)?
(a) Draw the signal flow graph and introduce the pipe-registers.
(b) In order to find the general case consider the adder with \(N=4\) and derive an expression for arbitrary \(N\).
(c) Assume that there are 3 such adders connected in series. Pipeline the circuit. What do you find out about the overall latency and the total number of pipe-registers? (keyword: skewing triangle)

3. Consider the Compare-select-unit (CSU) given in Fig. 3. Its output is given by the recurrence

\[ \gamma_{1,k+1} = \max(\gamma_{1,k} ; \gamma_{2,k}) \]  

\(\text{Figure 3: Compare-Select Recursion Loop}\)

(a) How can you implement a maximum circuit? (hint: There are basically two implementations with different feedback loops)
(b) Define the critical path at bit-level for both circuits.
(c) How can the c.p. made independent of the word width?
(d) (Appendix) How does the c.p. change when you implement an Add-Compare-Select circuit?

\[ \gamma_{1,k+1} = \max(\gamma_{1,k} + \lambda_{11,k} ; \gamma_{2,k}) \]  

4. A linear time-invariant discrete time system (LTI DT) can be expressed by its difference equation

\[ y_k = \sum_{j=0}^{Q} b_j x_{k-j} + \sum_{i=1}^{M} a_i y_{k-i} \]  

We assume for a fictive system \(Q = 0\) and \(b_j = 0\) for \(j > Q\). Furthermore the order of the system is set to 2, i.e. \(M = 2\).
(a) Draw the signal flow graph (e.g. direct form II realization) of the system
(b) Transform the recurrence such that the c.p. becomes as short as one add or one multiply. Decompose the signal flow graph into FIR and IIR part.
(c) Can you still apply the rules if you substitute the operations by \(\max\) and \(+\) using the operators \(\oplus\) and \(\ominus\)? In the first case we define \(\oplus = +\) and \(\ominus = x\), now \(\oplus = \max\) and \(\ominus = +\). (Recall the rules of the associative and distributive law.)

The system becomes \(y_k = (b_0 \ominus x_k) \oplus (a_1 \ominus y_{k-1}) \ominus (a_2 \ominus y_{k-2})\).

Draw the signal flow graph of such a system.