Networked Embedded Systems Lab
Prof. Marco Zimmerling

Introduction to Embedded Systems – WS 2022/23

Sample Solution to Exercise 7: Architecture Synthesis I

Task 1: Scheduling

Consider the sequence graph in Figure 1.

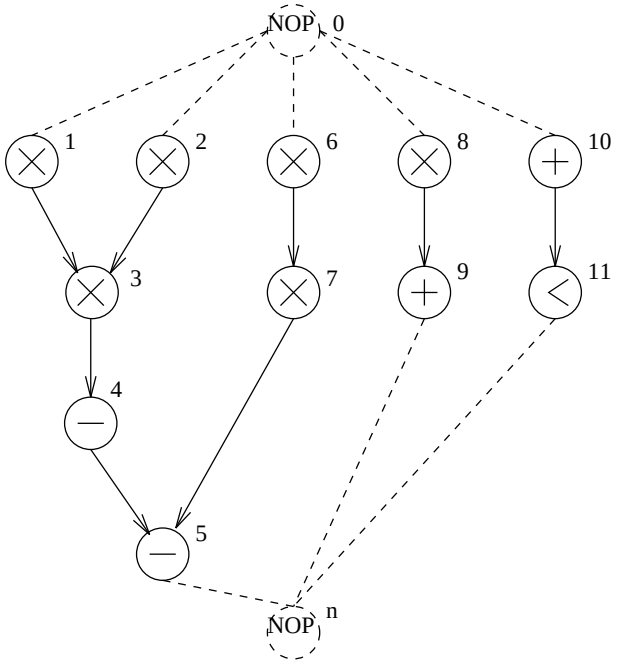

Figure 1: Sequence graph

All the operations are handled by the same resource type and have the same execution time:
$D_* = D_- = D_< = D_+ = 1$.

a) Set up a system of inequations which represent the constraints to valid schedules.

b) Set up an optimization model for the optimization of the latency L. Resource constraints need not be taken into account. (Hint: add an objective function to the system of inequations to get a linear program).

c) What is the minimal achievable latency

- if there is only one resource unit which handles all operations?
- if there are unlimited resource units?

Indicate valid starting times for the operations in both cases and check the validity of the schedules by means of the system of inequations determined previously.

Solution to Task 1:

System of inequations:

$$\begin{aligned}
\tau(v_3) &>= \tau(v_1) + 1 \\
\tau(v_3) &>= \tau(v_2) + 1 \\
\tau(v_4) &>= \tau(v_3) + 1 \\
\tau(v_5) &>= \tau(v_4) + 1 \\
\tau(v_5) &>= \tau(v_7) + 1 \\
\tau(v_7) &>= \tau(v_6) + 1 \\
\tau(v_9) &>= \tau(v_8) + 1 \\
\tau(v_{11}) &>= \tau(v_{10}) + 1
\end{aligned}$$

$$\begin{aligned}
\tau(v_1) &>= \tau(v_0) \\
\tau(v_2) &>= \tau(v_0) \\
\tau(v_6) &>= \tau(v_0) \\
\tau(v_8) &>= \tau(v_0) \\
\tau(v_{10}) &>= \tau(v_0)
\end{aligned}$$

$$\begin{aligned}
\tau(v_n) &>= \tau(v_5) + 1 \\
\tau(v_n) &>= \tau(v_9) + 1 \\
\tau(v_n) &>= \tau(v_{11}) + 1
\end{aligned}$$

Initial condition:

$$\tau(v_0) = 0$$

Objective function:

$$\min \quad \tau(v_n) - \tau(v_0)$$

Minimum latency and valid starting times:

- One resource: $L_{min} = 11$
- Unlimited resources: $L_{min} = 4$

1 resource: e.g.

$$\tau(v_1) = 0; \tau(v_2) = 1; \tau(v_3) = 2; \tau(v_4) = 3; \tau(v_6) = 4;$$
$$\tau(v_7) = 5; \tau(v_5) = 6; \tau(v_8) = 7; \tau(v_9) = 8; \tau(v_{10}) = 9;$$
$$\tau(v_{11}) = 10; \tau(v_n) = L = 11$$

Unlimited resources: e.g.

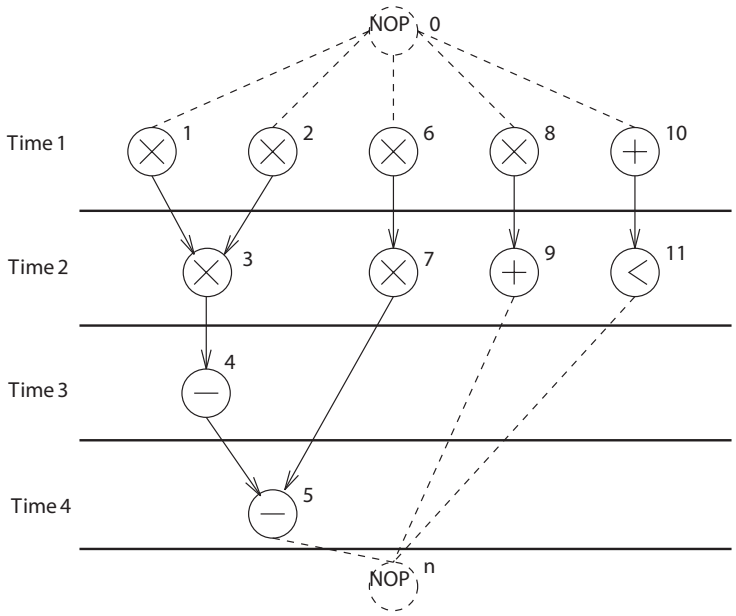

Figure 2: Starting times of operations for unlimited resources

Task 2: Design Space Exploration

Consider again the sequence graph and the specification of task 1. Assume that there is only one resource type which can compute all operations $(+, -, <, *)$ and has an area of 1. The cost of an implementation is given by the total required area. The goal is to find the Pareto-points of the design space which is given by the parameters cost and latency. The number of allocated resources is not yet fixed.

a) Compute a lower and an upper bound for the latency in order to limit the possible Pareto-points.

b) Find a lower and an upper bound for the cost in order to limit the possible Pareto-points.

c) Find all Pareto-points and represent them in a diagram.

Solution to Task 2:

Latency bounds: $4 \leq L \leq 11$. Cost bounds: $1 \leq c \leq 5$. Pareto-points: see Figure 3.

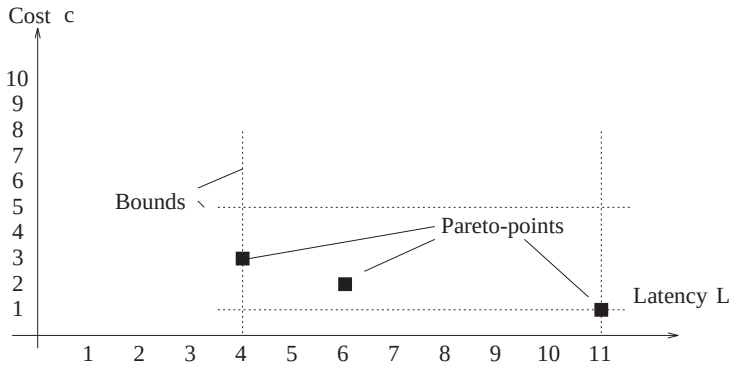

Figure 3: Pareto-points

Task 3: Marked Graphs

Consider the marked graph in Figure 4. The node labeled with + represents an addition of the two input values.

I) At the input *a* a sequence of numbers is read in, with *a(k)* representing the *k*-th number. Determine the outgoing sequence *b(k)* as function of the input values.

II) The initial mark with the value s is replaced by n marks $s_1, ..., s_n$. Determine a recursive formula for the output sequence *b(k)*.

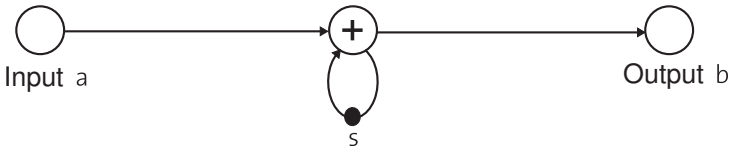

Figure 4: Marked Graph 1

Solution to Task 3:

I) $b(1) = a(1) + s$
$b(k) = a(k) + b(k-1)$ for $k > 1$
with s being the data value of the initial mark

II) For $n = 0$ the output sequence is empty.
For $n > 0$:
$b(k) = a(k) + s_k$ for $k <= n$
$b(k) = a(k) + b(k-n)$ for $k > n$
with $s_1, ..., s_n$ being the data values of the initial marks.

Task 4: List Scheduling

Given the sequence graph in Figure 5.

Suppose that two adders (r_1) and a multiplier (r_2) are available as resources. Addition takes one time unit and multiplication takes two time units. The first operation starts at $t = 0$ and the top node ('nop') is executed within zero time units. The priority is assigned for each operation as maximal distance to the bottom node ('nop').

a) Fill out Table 1 using the list scheduling algorithm. For a timestep t, $U_{t,k}$ denotes the set of operations that are ready to be scheduled on resource r_k (to be more specific, the set of operations that can be mapped on resource r_k and whose predecessors are all completed). $S_{t,k}$ denotes the set of operations that starts at time t on resource r_k, while $T_{t,k}$ is the set of operations in execution at time t on resource r_k.

b) What is the calculated latency?

c) Suppose that area costs for the adder and multiplier are 1 and 2 respectively. If it is allowed to spend additional hardware by 2 area units, which resource should be added to shorten the latency? Two adders or one multiplier? Explain why.

d) What is the new latency with the additional resource?

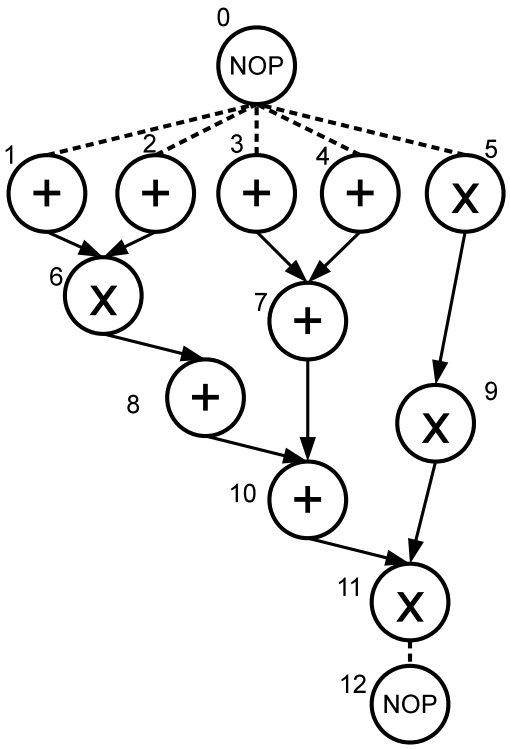

Figure 5: Sequence graph for Task 4

e) In case of unlimited hardware resources, what is the minimal latency?

Solution to Task 4:

a) See Table 1.

b) $L = 8$.

c) Multiplier because the critical path $(1 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 11)$ is not delayed by adder but multiplier.

d) $L = 7$.

e) $L = 7$.

t	k	$U_{t,k}$	$T_{t,k}$	$S_{t,k}$
0	r_1	$\nu_1, \nu_2, \nu_3, \nu_4$	—	ν_1, ν_2
	r_2	ν_5	—	ν_5
1	r_1	ν_3, ν_4	—	ν_3, ν_4
	r_2	ν_6	ν_5	—
2	r_1	ν_7	—	ν_7
	r_2	ν_6, ν_9	—	ν_6
3	r_1	—	—	—
	r_2	ν_9	ν_6	—
4	r_1	ν_8	—	ν_8
	r_2	ν_9	—	ν_9
5	r_1	ν_{10}	—	ν_{10}
	r_2	—	ν_9	—
6	r_1	—	—	—
	r_2	ν_{11}	—	ν_{11}
7	r_1	—	—	—
	r_2	—	ν_{11}	—
8	r_1	—	—	—
	r_2	—	—	—
9	r_1	—	—	—
	r_2	—	—	—
10	r_1	—	—	—
	r_2	—	—	—
11	r_1	—	—	—
	r_2	—	—	—

Table 1: Schedule for Task 4