Networked Embedded Systems Lab
Prof. Marco Zimmerling

Introduction to Embedded Systems – WS 2022/23

Sample Solution to Exercise 8: Architecture Synthesis II

Task 1: Scheduling with Pipeline Resources

Pipeline-resources process data in time intervals that are smaller than the actual execution time w. As soon as after the start of a task v_1 the so-called *pipeline-interval* PI has elapsed, the next task v_2 can be started on the same resource (see Figure 1). Non-pipeline-resources are a special case of pipeline-resources with $PI = w$.

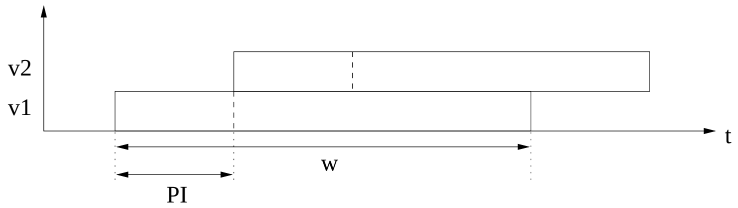

Figure 1: Tasks on pipeline-resource

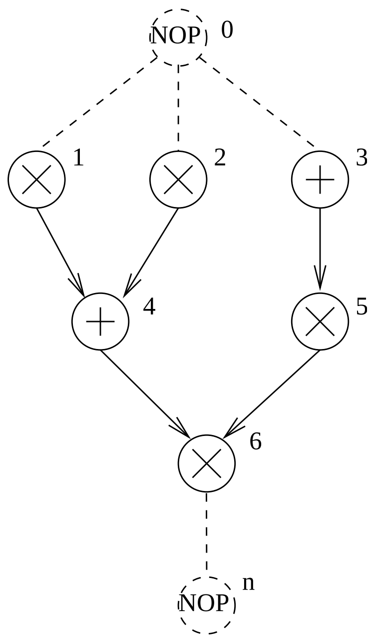

Figure 2: Sequence graph for Pipelining

a) Modify the LIST algorithm given in the lecture notes so that pipeline-resources are considered. Which step has to be reformulated and how? (Explain your answer!)

b) Perform the scheduling for the sequence graph given in Figure 2 using the modified algorithm. You can use Table 1. The multiplication (r_2) lasts 4 time units and the length of the pipeline-interval is 2 time units. The addition (r_1) lasts 2 time units and cannot be executed as pipeline-operation. 1 adder and 1 multiplier are available. Use the number of successor nodes as priority criterion. What is the resulting latency?

Solution to Task 1:

a) [...]
Determine candidates $U_{t,k}$ to be scheduled;
Determine set of occupied resources $O_{t,k}$*;*
Choose subset $S_t \subseteq U_{t,k}$ with maximal priority and $|S_{t,k}| + |O_{t,k}| \leq \alpha(v_k)$
[...]

$O_{t,k}$ is the set of resources of type k that are occupied in the time slot t and are not yet available for the following operation. On each of these resources exactly one operation is executed in a pipeline-interval.

b) The resulting schedule is shown in Table 1.
The resulting latency is 12.

t	k	$U_{t,k}$	$O_{t,k}$	$S_{t,k}$
0	r_1	v_3	—	v_3
	r_2	v_1, v_2	—	v_1
1	r_1	—	v_3	—
	r_2	v_2	v_1	—
2	r_1	—	—	—
	r_2	v_2, v_5	—	v_2
3	r_1	—	—	—
	r_2	v_5	v_2	—
4	r_1	—	—	—
	r_2	v_5	—	v_5
5	r_1	—	—	—
	r_2	—	v_5	—
6	r_1	v_4	—	v_4
	r_2	—	—	—
7	r_1	—	v_4	—
	r_2	—	—	—
8	r_1	—	—	—
	r_2	v_6	—	v_6
9	r_1	—	—	—
	r_2	—	v_6	—
10	r_1	—	—	—
	r_2	—	—	—
11	r_1	—	—	—
	r_2	—	—	—
12	r_1	—	—	—
	r_2	—	—	—

Table 1: Schedule for Task 1

Task 2: Integer Linear Programming

Given the sequence graph $G_S = (V_S, E_S)$ in Fig. 3.

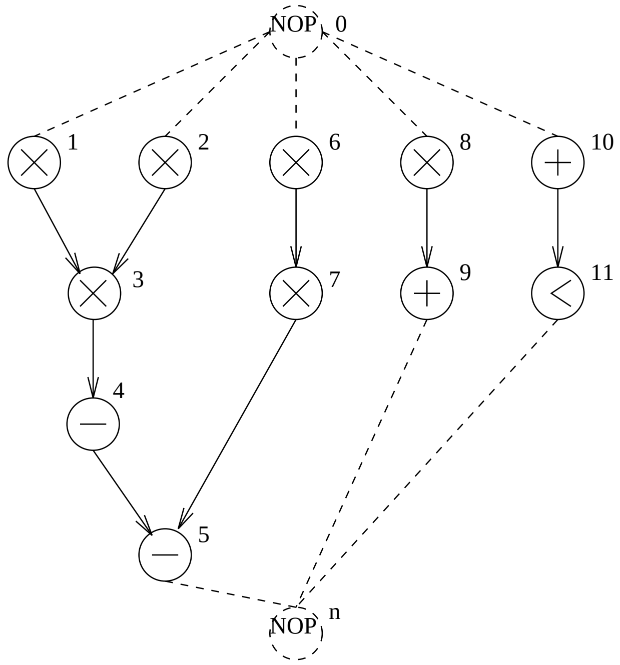

Figure 3: Sequence graph.

For the execution times of the operations assume: A multiplication operation (MULT) takes 2 time units and all other (ALU) operations take 1 time unit each. Two units of the resource type r_1 (multiplier) and two units of the resource type r_2 (ALU) are allocated.

(a) Apply the ASAP and ALAP algorithms to compute the earliest (l_i) and the latest (h_i) starting time of all operations $v_i \in V_s, i \in \{1, \ldots, 11\}$. For ALAP, assume the maximum latency $\overline{L} = 7$. Fill in the starting times in Table 2.

(b) Formulate the problem of latency minimization with restricted resources as an integer linear program (ILP). For this, you should introduce the binary variables $x_{i,t} \in \{0, 1\}\ \forall v_i \in V_S$ and $\forall t \in \{t \in \mathbb{Z} \mid l_i \leq t \leq h_i\}$. $\tau(v_i)$ is used to denote the starting time of operation $v_i \in V_S$ and $\alpha(r_i)$ with $r_i \in V_R = \{\mathrm{MULT}, \mathrm{ALU}\}$ denotes the number of allocated resource instances. Given the above notations, write down the following equations/inequations without using the $\sum$ symbol.

  (i) Express the objective function of the ILP
  (ii) Define $\tau(v_i)\ \forall i \in \{1, \ldots, 11\}$ as a function of $x_{i,t}$, where $l_1 \leq t \leq h_1$
  (iii) Express all data dependencies
  (iv) Express all resource limitations

(c) In an analogous manner try to formulate an ILP that solves the problem of cost minimization with latency limitation. Hint: We assume that the cost of a realization is the sum of the costs c of the multipliers with $c(r_1) = 2$ per allocated unit, and of the ALUs with $c(r_2) = 1$ per allocated unit. For the latency bound, we choose $\bar{L} = 6$.

Solution to Task 2:

(a) The starting times are listed in Table 2. The corresponding ASAP/ALAP schedules are depicted in Figure 4.

	l_i (ASAP)	h_i(ALAP)
v_1	1	2
v_2	1	2
v_3	3	4
v_4	5	6
v_5	6	7
v_6	1	3
v_7	3	5
v_8	1	5
v_9	3	7
v_{10}	1	6
v_{11}	2	7

Table 2: Earliest and latest starting times (Task 2a)

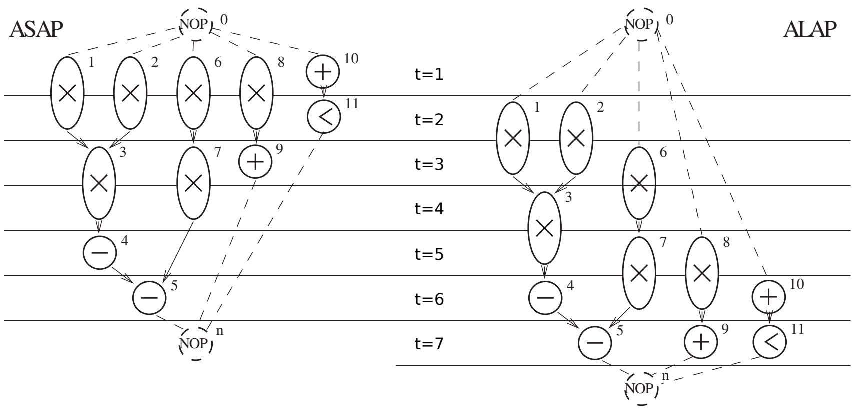

Figure 4: Schedule with ASAP and ALAP

(b) (i) Objective function:

$$\text{min.} \quad L = \tau(v_n) - \tau(v_0)$$

(ii) Introduction of binary variables:

$$
\begin{aligned}
x_{1,1} + x_{1,2} &= 1 & 1 \cdot x_{1,1} + 2 \cdot x_{1,2} &= \tau(v_1) \\
x_{2,1} + x_{2,2} &= 1 & 1 \cdot x_{2,1} + 2 \cdot x_{2,2} &= \tau(v_2) \\
x_{3,3} + x_{3,4} &= 1 & 3 \cdot x_{3,3} + 4 \cdot x_{3,4} &= \tau(v_3) \\
x_{4,5} + x_{4,6} &= 1 & 5 \cdot x_{4,5} + 6 \cdot x_{4,6} &= \tau(v_4) \\
x_{5,6} + x_{5,7} &= 1 & 6 \cdot x_{5,6} + 7 \cdot x_{5,7} &= \tau(v_5) \\
x_{6,1} + x_{6,2} + x_{6,3} &= 1 & 1 \cdot x_{6,1} + 2 \cdot x_{6,2} + 3 \cdot x_{6,3} &= \tau(v_6) \\
x_{7,3} + x_{7,4} + x_{7,5} &= 1 & 3 \cdot x_{7,3} + 4 \cdot x_{7,4} + 5 \cdot x_{7,5} &= \tau(v_7) \\
x_{8,1} + \ldots + x_{8,5} &= 1 & 1 \cdot x_{8,1} + \ldots + 5 \cdot x_{8,5} &= \tau(v_8) \\
x_{9,3} + \ldots + x_{9,7} &= 1 & 3 \cdot x_{9,3} + \ldots + 7 \cdot x_{9,7} &= \tau(v_9) \\
x_{10,1} + \ldots + x_{10,6} &= 1 & 1 \cdot x_{10,1} + \ldots + 6 \cdot x_{10,6} &= \tau(v_{10}) \\
x_{11,2} + \ldots + x_{11,7} &= 1 & 2 \cdot x_{11,2} + \ldots + 7 \cdot x_{11,7} &= \tau(v_{11})
\end{aligned}
$$

(iii) Data dependencies:

$$\begin{aligned} \tau(v_3) - \tau(v_1) &\geq 2 \qquad & \tau(v_3) - \tau(v_2) &\geq 2 \\ \tau(v_4) - \tau(v_3) &\geq 2 & \tau(v_5) - \tau(v_4) &\geq 1 \\ \tau(v_7) - \tau(v_6) &\geq 2 & \tau(v_5) - \tau(v_7) &\geq 2 \\ \tau(v_9) - \tau(v_8) &\geq 2 & \tau(v_{11}) - \tau(v_{10}) &\geq 1 \\ \tau(v_n) - \tau(v_5) &\geq 1 & \tau(v_n) - \tau(v_9) &\geq 1 \\ \tau(v_n) - \tau(v_{11}) &\geq 1 \end{aligned}$$

$$\tau(v_1), \tau(v_2), \tau(v_6), \tau(v_8), \tau(v_{10}) \geq \tau(v_0) \geq 1$$

(iv) Resource limitations:

$t = 1$:

$$x_{1,1} + x_{2,1} + x_{6,1} + x_{8,1} \leq 2$$
$$x_{10,1} \leq 2$$

$t = 2$:

$$x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2} + x_{6,1} + x_{6,2} + x_{8,1} + x_{8,2} \leq 2$$
$$x_{10,2} + x_{11,2} \leq 2$$

$t = 3$:

$$x_{1,2} + x_{2,2} + x_{6,2} + x_{6,3} + x_{8,2} + x_{8,3} + x_{3,3} + x_{7,3} \leq 2$$
$$x_{10,3} + x_{11,3} + x_{9,3} \leq 2$$

$t = 4$:

$$x_{6,3} + x_{8,3} + x_{8,4} + x_{3,3} + x_{3,4} + x_{7,3} + x_{7,4} \leq 2$$
$$x_{10,4} + x_{11,4} + x_{9,4} \leq 2$$

$t = 5$:

$$x_{8,4} + x_{8,5} + x_{3,4} + x_{7,4} + x_{7,5} \leq 2$$
$$x_{10,5} + x_{11,5} + x_{9,5} + x_{4,5} \leq 2$$

$t = 6$:

$$x_{8,5} + x_{7,5} \leq 2$$
$$x_{10,6} + x_{11,6} + x_{9,6} + x_{4,6} + x_{5,6} \leq 2$$

$t = 7$:

$$(0 \leq 2)$$
$$x_{11,7} + x_{9,7} + x_{5,7} \leq 2$$

(c) Restating the resource limitations, and introducing additional variables:

$t = 1$:

$$x_{1,1} + x_{2,1} + x_{6,1} + x_{8,1} - \alpha(r_1) \leq 0$$
$$x_{10,1} - \alpha(r_2) \leq 0$$

[...]

Latency limitations:

$$L = \tau(v_n) - \tau(v_0) \leq \bar{L} = 6$$

New objective function:

$$\text{min.} \quad C = \alpha(r_1) \cdot c(r_1) + \alpha(r_2) \cdot c(r_2) = 2 \cdot \alpha(r_1) + \alpha(r_2)$$

Task 3: Iterative Algorithms

Please answer the following questions considering the given video codec application specified as a marked graph in Figure 5.

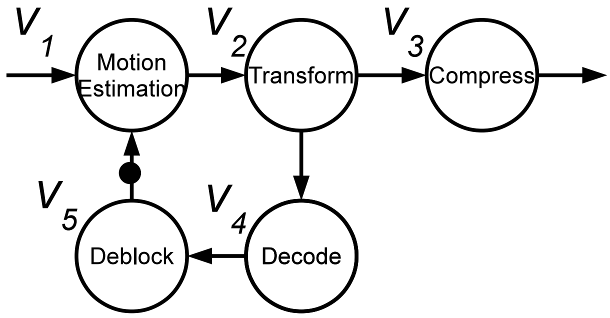

Figure 5: Video codec marked graph representation

	ν_1	ν_2	ν_3	ν_4	ν_5
$w(\nu_i)$	10	10	10	5	5

Table 3: Execution time of each function

(a) Formulate all existing dependencies in Figure 5 from ν_i to ν_j in the form of

$$\tau(\nu_j) - \tau(\nu_i) \geq w(\nu_i) - d_{ij} \cdot P,$$

where P is the minimum iteration interval. The execution time of each function is listed in Table 3.

(b) Assuming unlimited resources and only one token on the edge between ν_5 and ν_1, determine the minimum iteration interval P and the latency L. To justify your answer, draw the scheduling on the timeline given in Figure 6 with the dependency from ν_5 to ν_1 highlighted.

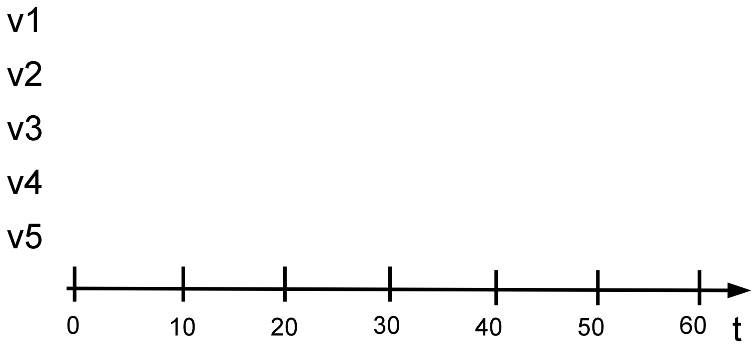

Figure 6: Scheduling result of the video codec

(c) The motion estimation function (ν_1) uses the result of the previous frame (See the dependency between ν_1 and ν_5). Let us now suppose that any arbitrary number of tokens can be inserted to reduce P using functional pipelining. Then, determine the minimum number of tokens that should be added on the edge $\nu_5 \rightarrow \nu_1$ to achieve $P = 10$? To justify your answer, draw the pipelined scheduling on the timeline given in Figure 7 with the dependency from ν_5 to ν_1 highlighted and calculate the latency L of the schedule.

Solution to Task 3:

(a) Dependencies:
$\tau(\nu_2) - \tau(\nu_1) \geq 10$
$\tau(\nu_3) - \tau(\nu_2) \geq 10$
$\tau(\nu_4) - \tau(\nu_2) \geq 10$

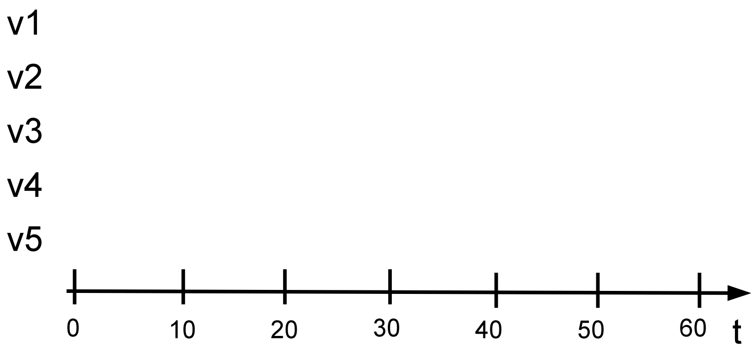

Figure 7: Pipelined scheduling result of the video codec

$\tau(\nu_5) - \tau(\nu_4) \geq 5$
$\tau(\nu_1) - \tau(\nu_5) \geq 5 - 1 \cdot P$

(b) We solve the system of inequalities of 3a) for P.
$\Rightarrow P_{min} = 30$
$L = 30$

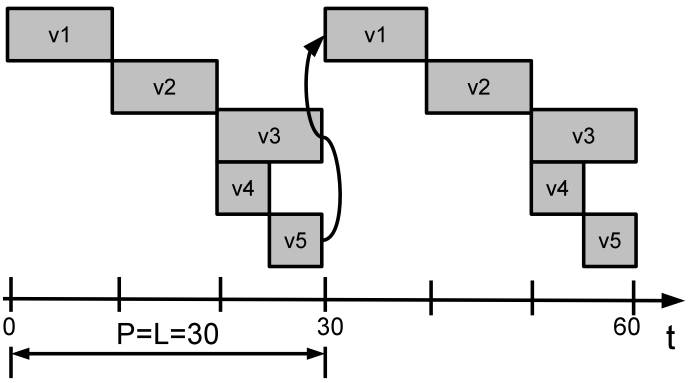

Figure 8: Scheduling result of the video codec

(c) Now the iteration interval P is given ($P = 10$) and we are looking for the number of tokens n. Therefore, we replace the last inequation in 3a) by $\tau(\nu_1) - \tau(\nu_5) \geq 5 - n \cdot 10$ and solve the new set of inequations for n.
$\Rightarrow n_{min} = 3$
We have to add at least 2 tokens on the edge between ν_5 and ν_1.
$L = 30$

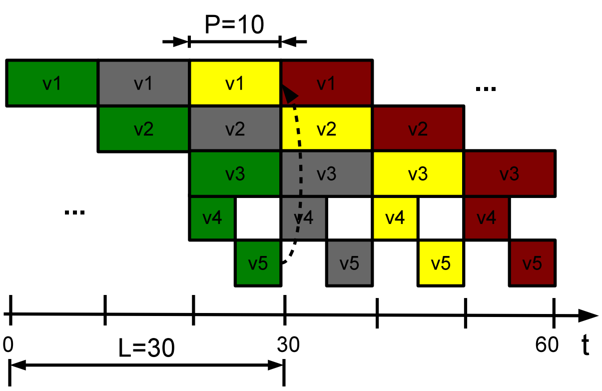

Figure 9: Pipelined scheduling result of the video codec