Architecture Synthesis I

Scheduling with Pipeline Resources, Integer Linear Programming, Iterative Algorithms

Exercise class 10

Presenter:
Jürgen Mattheis

In cooperation with:
Pascal Walter

Based on the lecture of:
Marco Zimmerling

January 29, 2023

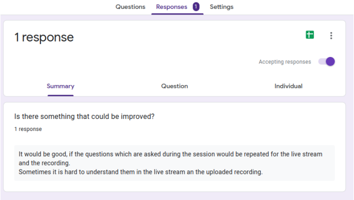
University of Freiburg, Chair for Embedded Systems

Gliederung

Organisation

Organisation I

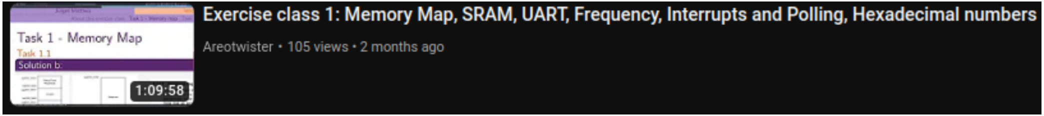
- feedback for lecture until February 29
- feedback for me: `https://forms.gle/f3YN8EFrZ1vsfPoC6`



- feedback from students in the last exercise class: don't overexplain tasks at the beginning so there's enough time for the tasks at the end

Organisation I

- the exercise class videos on youtube do now have a title containing the topics covered in the exercises of the corresponding exercise class

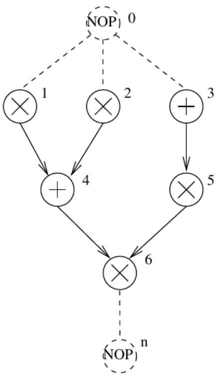


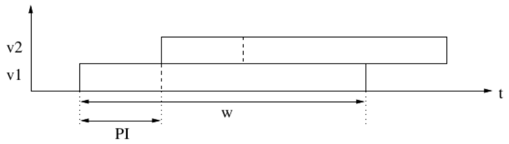
Task 1

Task 1 I

Scheduling with Pipeline Resources

Task 1.1:





Task 1

Scheduling with Pipeline Resources

Solution 1.1:

```
LIST(G_S(V_S, E_S), G_R(V_R, E_R), α, β, priorities){
  t = 1;
  REPEAT {
    FORALL v_k ∈ V_T {
      Determine candidates U_{t,k} to be scheduled;
      Determine set of occupied resources O_{t,k};
      Choose subset S_t ⊆ U_{t,k} with maximal priority
        and |S_{t,k}| + |O_{t,k}| ≤ α(v_k);
      τ(v_i) = t ∀v_i ∈ S_k;   }
    t = t + 1;
  } UNTIL (v_n planned)
  RETURN (τ);  }
```

Task 1

Scheduling with Pipeline Resources

Solution 1.1:

- *$O_{t,k}$ 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.*
- $O_{t,k} = \{v_s : \beta(v_s) = v_t \wedge \tau(v_s) < t < \tau(v_s) + PI\}$ *instead of*
  $T_{t,k} = \{v_s : \beta(v_s) = v_t \wedge \tau(v_s) < t < \tau(v_s) + w(v_s, v_t)\}$

Task 1

Scheduling with Pipeline Resources

Solution 1.1:

- *without pipelining*:

t	k	$U_{t,k}$	$T_{t,k}$	$S_{t,k}$
0	r_1	v3	-	v3
	r_2	v1 v2	-	v1
1	r_1	-	v3	-
	r_2	v2	v1	-
2	r_1	-	-	-
	r_2	v2 v5	v1	-
3	r_1	-	-	-
	r_2	v2 v5	v1	-
4	r_1	-	-	-
	r_2	v5	-	v2
5	r_1	-	-	-
	r_2	v5	v2	-
6	r_1	-	-	-
	r_2	v5	v2	-
7	r_1	-	-	-
	r_2	v5	v2	-
8	r_1	v4	-	v4
	r_2	v5	-	v5
9	r_1	-	v4	-
	r_2	-	v5	-
10	r_1	-	-	-
	r_2	-	v5	-
11	r_1	-	-	-
	r_2	-	v5	-
12	r_1	-	-	-
	r_2	v6	-	v6
13	r_1	-	-	-
	r_2	-	v6	-
14	r_1	-	-	-
	r_2	-	v6	-
15	r_1	-	-	-
	r_2	-	v6	-
15	r_1	-	-	-
	r_2	-	-	-

Task 1

Scheduling with Pipeline Resources

Solution 1.1:

t	k	$U_{t,k}$	$O_{t,k}$	$S_{t,k}$
0	r_1			
	r_2			
1	r_1			
	r_2			
2	r_1			
	r_2			
3	r_1			
	r_2			
4	r_1			
	r_2			
5	r_1			
	r_2			
6	r_1			
	r_2			
7	r_1			
	r_2			
8	r_1			
	r_2			
9	r_1			
	r_2			
10	r_1			
	r_2			
11	r_1			
	r_2			
12	r_1			
	r_2			

Task 1

Scheduling with Pipeline Resources

Solution 1.1:

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	—	—	—

Solution 1.1:

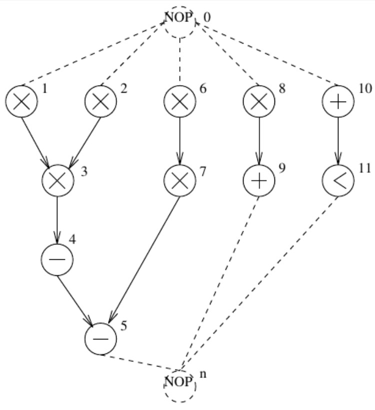
- *the resulting latency is* 12

Task 2

Task 2 I

Integer Linear Programming

Task 2.1:



- resource type r_1: multiplication operation takes 2 time units and 2 units of this resource type are allocated
- resource type r_2: all other ALU Operations take 1 time unit and 2 units of this resource type are allocated

Task 2 II

Integer Linear Programming

Requirements 2.1:

ASAP($G_S(V_S, E_S), w$) {
  $\tau(v_0) = 1$;
  REPEAT {
    Determine v_i whose predec. are planed;
    $\tau(v_i) = \max\{\tau(v_j) + w(v_j)\ \forall (v_j, v_i) \in E_S\}$
  } UNTIL (v_n is planned);
  RETURN (τ);
}

Task 2 III

Integer Linear Programming

Requirements 2.1:

ALAP($G_S(V_S, E_S), w, L_{max}$) {
  $\tau(v_n) = L_{max} + 1$;
  REPEAT {
    Determine v_i whose succ. are planed;
    $\tau(v_i) = \min\{\tau(v_j)\ \forall (v_i, v_j) \in E_S\} - w(v_i)$
  } UNTIL (v_0 is planned);
  RETURN (τ);
}

Task 2 IV

Integer Linear Programming

Solution 2.1:

	l_i (ASAP)	h_i(ALAP)
v_1		
v_2		
v_3		
v_4		
v_5		
v_6		
v_7		
v_8		
v_9		
v_{10}		
v_{11}		

Task 2 V

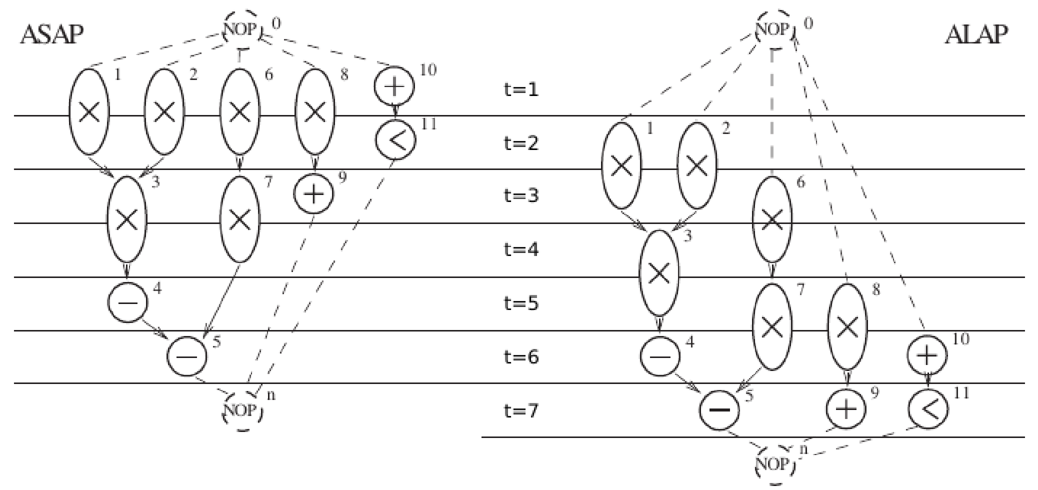
Integer Linear Programming

Solution 2.1:

	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

Task 2 VI

Solution 2.2:



Task 2 I

Integer Linear Programming

Solution 2.2:

- *(i) Objective function:*

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

- *(ii) Introduction of binary variables:*

$$\begin{aligned}
x_{1,1} + x_{1,2} &= 1 & 1 \cdot x_{1,1} + 2 \cdot x_{1,2} &= \tau(v_1) \\
x_{2,1} + x_{2,2} &= 1 & 1 \cdot x_{2,1} + 2 \cdot x_{2,2} &= \tau(v_2) \\
x_{3,3} + x_{3,4} &= 1 & 3 \cdot x_{3,3} + 4 \cdot x_{3,4} &= \tau(v_3) \\
x_{4,5} + x_{4,6} &= 1 & 5 \cdot x_{4,5} + 6 \cdot x_{4,6} &= \tau(v_4) \\
x_{5,6} + x_{5,7} &= 1 & 6 \cdot x_{5,6} + 7 \cdot x_{5,7} &= \tau(v_5)
\end{aligned}$$

Task 2 II

Integer Linear Programming

Solution 2.2:

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

Task 2 III

Integer Linear Programming

Solution 2.2:

- *(iii) Data dependencies:*

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

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

Task 2 IV

Integer Linear Programming

Solution 2.2:

- *(iv) Resource limitations:*
  - $t = 1$:

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

  - $t = 2$:

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

  - $t = 3$:

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

Task 2 V

Solution 2.2:

- $t = 4$:

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

- $t = 5$

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

- $t = 6$:

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

- $t = 7$

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

Integer Linear Programming I

Solution 2.3:

- *Restating the resource limitations, and introducing additional variables:*
  - $t = 1$:

$$x_{1,1} + x_{2,1} + x_{6,1} + x_{8,1} \leq \alpha(r_1)$$
$$x_{10,1} \leq \alpha(r_2)$$
$$[...]$$

  - *Latency limitations:*

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

  - *New objective function:*

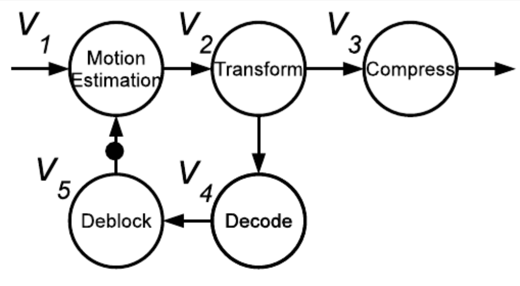
$$min\ C = \alpha(r_1) \cdot c(r_1) + \alpha(r_2) \cdot c(r_2) = 2 \cdot \alpha(r_1) + \alpha(r_2)$$

Task 3

Tasks 3 I

Iterative Algorithms

Task 3.1:



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

Tasks 3 II

Iterative Algorithms

Solution 3.1:

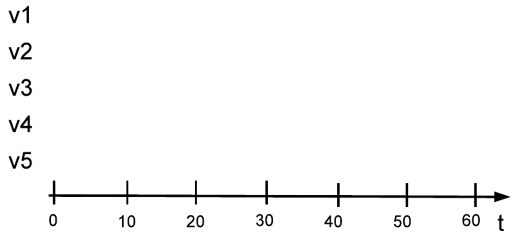
- $\tau(\nu_2) - \tau(\nu_1) \geq 10$
- $\tau(\nu_3) - \tau(\nu_2) \geq 10$
- $\tau(\nu_4) - \tau(\nu_3) \geq 10$
- $\tau(\nu_5) - \tau(\nu_4) \geq 5$
- $\tau(\nu_1) - \tau(\nu_5) \geq 5 - 1 \cdot P$

Tasks 3 III

Iterative Algorithms

Solution 3.2:

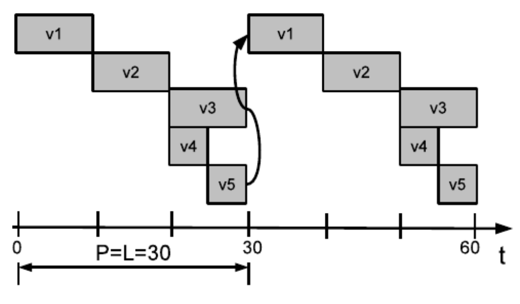
- *solve system of inequalities for P:*
  - $0 - 25 \geq 5 - P \Leftrightarrow P_{min} = 30$
  - $L = 30$



Tasks 3 IV

Iterative Algorithms

Solution 3.2:

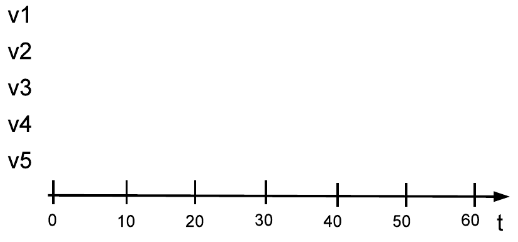


Tasks 3 V

Iterative Algorithms

Solution 3.3:

- $\tau(v_1) - \tau(v_5) \geq 5 - n \cdot 10 \Leftrightarrow n_{min} = 3$
- *we have to add 2 more tokens on the edge between v_5 and v_1*
- $L = 30$



Tasks 3 VI

Iterative Algorithms

Solution 3.3: