



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

Institut für Technische Informatik  
und Kommunikationsnetze  
Computer Engineering and Networks Laboratory



Exam Autumn 2017– Sample solution

# Hardware/Software Codesign

Prof. L. Thiele

**Note:**

The given solution is only a proposal. For correctness, completeness, or understandability, no responsibility is taken.

**Task 1 : Models of Computation**

(maximal 38 points)

**1.1: State Charts**

(maximal 12 points)

Observe the state chart in Figure 1. The following events can trigger a state change: {'red', 'blue', 'powerOn', 'powerOff', 'plus', 'minus', 'sO', 'sA', 'sB'}. 'vA' and 'vB' are two internal variables. For this state chart answer the questions below.

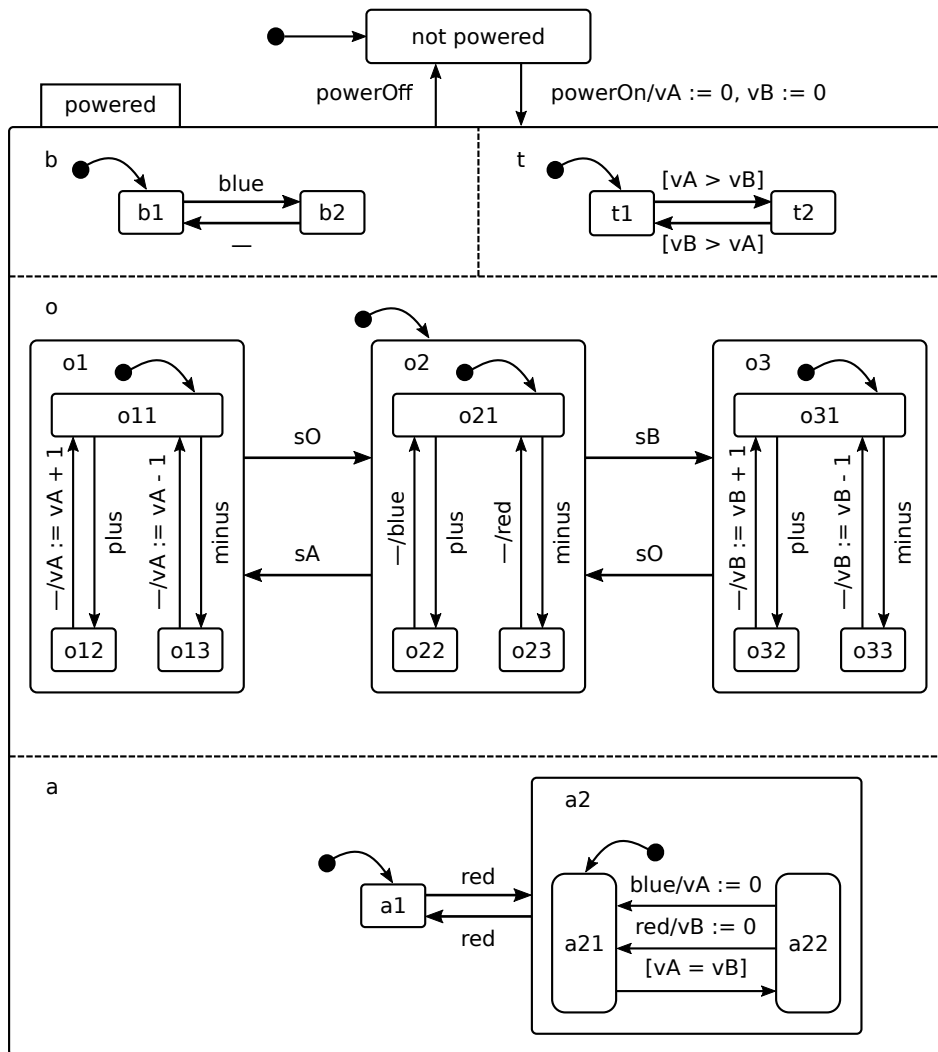


Figure 1: A state chart.

- (a) (3 points) In how many different states can the system be? For this question, exclude the variables 'vA' and 'vB'.

**Sample solution:**

There are 109 states in total:  $1 + 2 \cdot 2 \cdot 3 \cdot 9$

- (b) (9 points) Determine the sequence of states, starting from the state 'not powered', if the following sequence of external events occurs: 'powerOn', 'red', 'sA', 'plus', 'sO' and 'plus'. Fill out Table 1 by specifying only the **stable** state reached after each external event.

**Sample solution:**

event	b	o	a	t
'powerOn'	b1	o21	a1	t1
'red'	b1	o21	a22	t1
'sA'	b1	o11	a22	t1
'plus'	b1	o11	a22	t2
'sO'	b1	o21	a22	t2
'plus'	b1	o21	a22	t2

Table 1: System state after external events.

**1.2: Dataflow Languages**

(maximal 18 points)

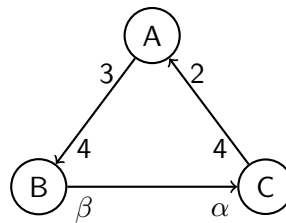
A system modeled by an SDF graph has three processes named 'A', 'B' and 'C', and has the following topology matrix:

$$\begin{pmatrix} 3 & -4 & 0 \\ 0 & \beta & -\alpha \\ -2 & 0 & 4 \end{pmatrix} \quad (1)$$

Suppose that  $\alpha$  and  $\beta$  are positive integers, and initial tokens exist only on the edge between nodes 'A' and 'C'.

- (a) (3 points) Draw the SDF graph and attach labels to the heads and tails of edges that represent the number of tokens consumed and produced at each firing of the corresponding process.

**Sample solution:**



- (b) (9 points) Given that the system has sufficient number of initial tokens, determine minimal integer values for the parameters  $\alpha$  and  $\beta$  such that the system has a periodic schedule. Determine the relative execution rates of the processes.

**Sample solution:**

For a periodic schedule to exist, the rank of topology matrix must be 2.

Observe, from the topology matrix, that  $2 \cdot \text{Row1} + 3 \cdot \text{Row3} = (0 \ -8 \ 12)$ . From this we can conclude that the minimum integer values are  $\beta = 2, \alpha = 3$

The relative firing rates of A, B, and C are 4, 3, and 2 respectively.

- (c) (6 points) Determine a periodic schedule that requires the minimal number of initial tokens on the edge between nodes 'A' and 'C'.

**Sample solution:**

The schedule  $4A3B2C$  requires six initial tokens between nodes 'A' and 'C'.

**1.3: KPNs and SDFs**

(maximal 8 points)

- (a) (2 points) A process is described below. Based on the description, argue shortly whether this is a Kahn process or not.

The process 'MERGE' has two input channels and one output channel. When a token appears on any of the input ports, and no token appears at the same time on the other port, it is forwarded to the output channel immediately. If tokens appear on the two input ports simultaneously, they are forwarded to the output immediately, but one predetermined input port's token is always forwarded first. This way, the two input channels are merged into one output.

**Sample solution:**

---

This is not a Kahn process: it is random (non-determinate), as the output depends on the time events arrive.

---

- (b) (2 points) A process is described below. Based on the description, argue shortly whether this is a Kahn process or not.

The process 'SPLIT' has one input channel and two output channels. Tokens appearing on the input port are forwarded immediately to one of the output channels in an alternating fashion (i.e. first to one, then to the other, and so on). This way, the input channel is split into two output channels, with each of the output channels receiving half of the events.

**Sample solution:**

---

This is a Kahn process: it is determinate, and can be described with imperative code with destructive and blocking read, and non-blocking write.

---

- (c) (2 points) A function is given below. It uses the function `send(int, OutChannel)` to place one token on the specified communication channel. Values `a` and `b` are integers that were read from input channels, while `ch` is an output channel. Based on the code given below, argue shortly whether this could be the fire function of an SDF actor.

```
function fire() {
    if (a < b) {
        send(a + b, ch);
    }
    else {
        send(a - b, ch);
    }
}
```

**Sample solution:**

---

This can be the fire function of a SDF actor, as it always produces the same number of tokens when called.

---

- (d) (2 points) A function is given below. It uses the function `send(int, OutChannel)` to place one token on the specified communication channel. Values `a` and `b` are integers that were read from input channels, while `ch` is an output channel. Based on the code given below, argue shortly whether this could be the fire function of an SDF actor.

```
function fire() {
    if (a < b) {
        printf("Warning: The result is not positive");
    }
    else {
        send(a - b, ch);
    }
}
```

**Sample solution:**

---

This can not be the fire function of a SDF actor, as it does not produce the same number of tokens each time it is called.

---

**Task 2 : Performance Analysis**

(maximal 36 points)

Determine the Best Case Execution Time (BCET) of the following program, which is written in pseudo code:

```
j = 0;
a = k + 2;

if (i < 10) {
  if (b)
    j = k + a;
  else {
    b = TRUE;
    c = 2 * a + c;
  }
  i = i + 1;
}
k = i + j;
```

The underlying machine executing the program is specified as follows:

- the processor has no pipeline.
- the processor has no registers, i.e. all variables are stored in memory.
- in an assignment, predicate evaluation, or arithmetic operation, variables are accessed in order from right to left e.g.

$a = b + c$

variables are accessed in the order c, b, a.

- consider only the data cache.
- reads and writes from/to a variable are treated the same way. If the required data is in the cache, there is a cache HIT and it takes **1 cycle** to read/write the data. If the data is not in the cache, there is a cache MISS and it takes **100 cycles** to read/write the data.
- execution model is very simplified. No temporary results are considered. Execution times of all processor operations are 1 cycle. In addition, the time for loading and storing a variable needs to be taken into account.



- the data cache is 2-way set-associative. The cache can store a total of 4 data blocks in 2 sets. Each cache block can contain at most one variable. The variables are mapped to sets in the following way:

Variable	Set
a,b,c	1
i,j,k	2

- the cache uses LRU replacement policy within each set once it is full.

Initial conditions:  $a, c, i, j, k \in (-100, 100)$ ,  $b \in \{\text{TRUE}, \text{FALSE}\}$  and cache state is  $((\underbrace{a, -}_{\text{set 1}}, \underbrace{-, -}_{\text{set 2}})$ ,

i.e. set 1 has variable  $a$  in its most recently used block, set 2 is empty.

- (a) (6 points) Determine the basic blocks of the program and draw the corresponding control flow graph.

**Solution:**

See Figure 2.

- (b) (20 points) Use cache MAY analysis to determine the best-case execution time for each basic block.

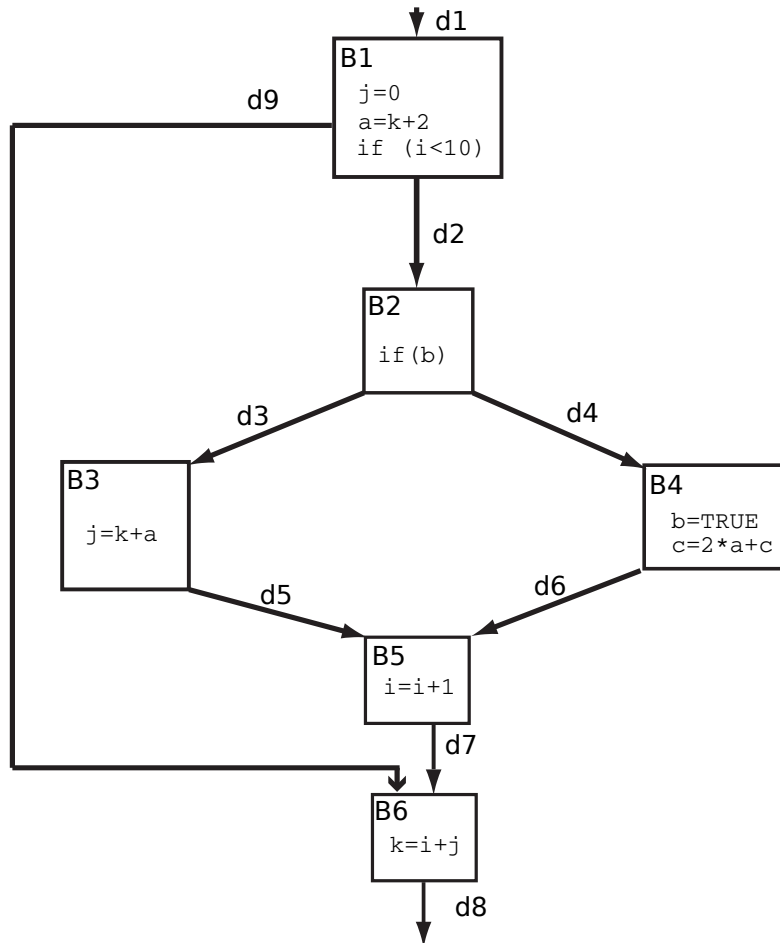


Figure 2: Basic blocks of the program.

**Solution:**

Use cache MAY analysis to determine the BCET of each basic block.

H for hit, M for miss,  
 ci is BCET in cycles for the i-th basic block  
 Initial cache state: ((a,-),(-,-))

B1:  
 j = 0  
 M, ((a,-),(j,-))

```
a = k + 2
  M, ((a,-),(k,j))
  H, ((a,-),(k,j))
if (i<10)
  M, ((a,-),(i,k))
C1: 303
```

```
B2:
  if(b)
    M, ((b,a),(i,k))
  c2: 101
```

```
B3:
  j = k + a
  H, ((a,b),(i,k))
  H, ((a,b),(k,i))
  M, ((a,b),(j,k))
  c3: 103
```

```
B4:
  b=true
  H, ((b,a),(i,k))
  c=2*a+c
  M, ((c,b),(i,k))
  M, ((a,c),(i,k))
  H, ((c,a),(i,k))
```

```
c4: 203
```

```
Join of B3 and B4: (({a,c},{b}),(j,i,{k}))
```

```
B5:
  i = i+1
  H, (({a,c},{b}),(i,{j,k}))
  H, (({a,c},{b}),(i,{j,k}))
  c5: 3
```

```
Join of B1 and B5: (({a,c},{b}),(i,{j,k}))
```

```
B6:
```

$k = i + j$   
H:  $((\{a, c\}, \{b\}), (\{j\}, \{i\}))$   
H:  $((\{a, c\}, \{b\}), (\{i\}, \{j\}))$   
M:  $((\{a, c\}, \{b\}), (\{k\}, \{i\}))$   
c6: 103

(c) (10 points) Write down the structural constraints and the ILP to determine the BCET of one program execution.

**Solution:**

Flow variables  $d_i$  are defined in Figure 2.

Structural constraints:

$$d_1 = d_2 + d_9 = x_1$$

$$d_2 = d_3 + d_4 = x_2$$

$$d_3 = d_5 = x_3$$

$$d_4 = d_6 = x_4$$

$$d_5 + d_6 = d_7 = x_5$$

$$d_9 + d_7 = d_8 = x_6$$

The program is executed once:  $d_1 = 1$ .

Objective function:

$$BCET = \min\left\{\sum_{i=1}^6 c_i * x_i\right\}$$

**Task 3 : Optimization, Mapping, Partitioning, and Design Space Exploration**  
(maximal 46 points)

**3.1: Multi-Criteria Optimization**

(maximal 11 points)

Table 3 shows a design space with 8 points in 3 dimensions. All objectives, the cost  $C$ , the power dissipation  $P$ , and the execution time  $T$  are to be minimized.

Point	Cost [CHF]	Power [mW]	Time [ms]
P1	20	450	800
P2	70	300	500
P3	80	150	300
P4	40	100	400
P5	90	50	200
P6	50	300	700
P7	40	350	900
P8	70	200	500

Table 2: Three dimensional design points.

(a) (5 points) Identify the points from Table 3 which belong to the Pareto front.

**Sample solution:**

Here, we will analyze each point, and determine whether there is any other point that dominates it.

Point	Dominated by
P1	None
P2	P4
P3	None
P4	None
P5	None
P6	P4
P7	P4
P8	P4

Table 3: Three dimensional design points.

Since Point 1 (20,450,800), Point 3 (80,150,300), Point 4 (40,100,400) and Point 5 (90,50,200) are not dominated by other point, they are Pareto points. Since they are different in all dimensions, they are all in the Pareto front. These points are marked red in figure 3. **Drawing this figure is not required in the exam.**

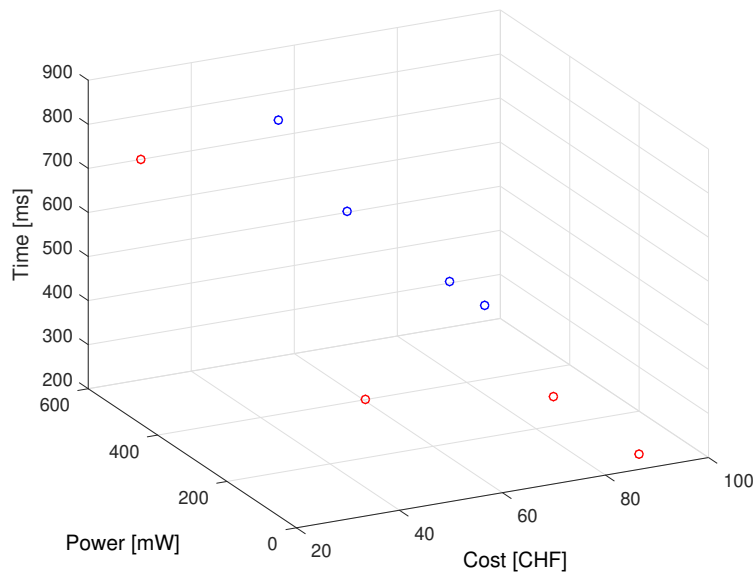


Figure 3: 3D design points. Red points belong to the Pareto front.

- (b) (6 points) Consider only the Cost and Power dimensions. Draw the points in Fig. 4. Given the reference point  $R(100[CHF], 500[mW])$ , find the subset with two points that has the *largest* hypervolume and determine the value of this hypervolume.

**Sample solution:**

The hypervolume is obtained from the figure, and is 51 units (shaded gray in figure 4).

Removing point P5, will cause the hypervolume to be the largest: 50 units, or  $50 * 10 CHF * 50 mW = 25\ 000\ mW\ CHF$ . Therefore, the subset is  $\{P1, P4\}$ , or  $\{(40, 100), (20, 450)\}$ .

**Shading the hypervolume in the figure is not required in the exam.**

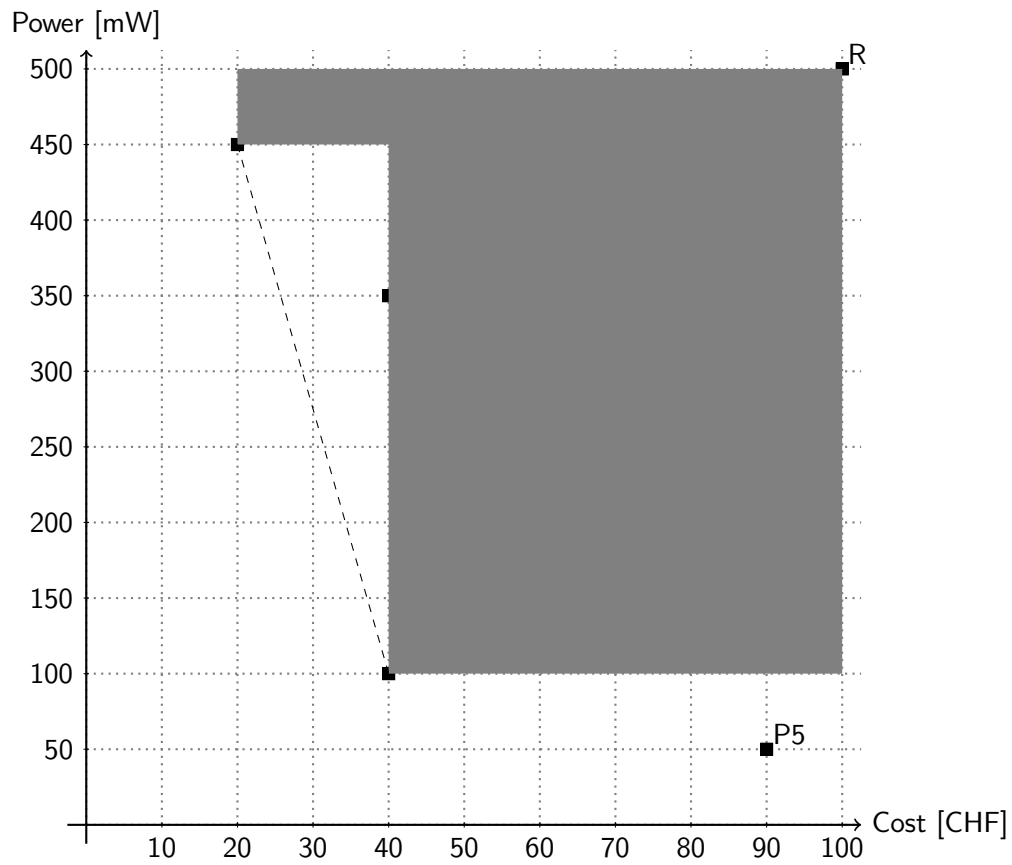


Figure 4: Two-dimensional design space.

**3.2: Design Space Exploration**

(maximal 19 points)

Figure 5 shows a specification at the system level consisting of a synchronous data flow (SDF) graph and an architecture template. Table 4 presents the execution time of the different actors on the available processors. We have one instance of each processor available. Each token has a size of 100 bit, and the bus has a bandwidth of 5000 bit/s. Communication between tasks bound to the same processor does not require any time. The SDF graph is executed 10 times per second.

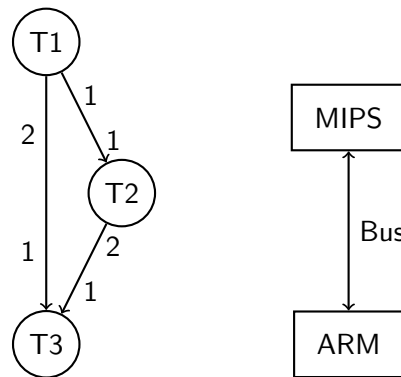


Figure 5: System-level specification.

Table 4: Execution times for tasks.

processor	execution time [ms]		
	T1	T2	T3
MIPS	5	30	-
ARM	50	10	20

- (a) (4 points) Construct the specification graph. Your graph should include the problem graph with communication nodes, the architecture graph, and all valid bindings.

**Sample solution:**

See Figure 6.

- (b) (3 points) Determine the minimal number of activations by each actor that are needed to complete one SDF graph execution.

**Sample solution:**

A valid schedule for the SDF graph is (T1-T2-T3-T3)\*. Hence, to complete one iteration, T1 and T2 require one activation, while T3 requires two.



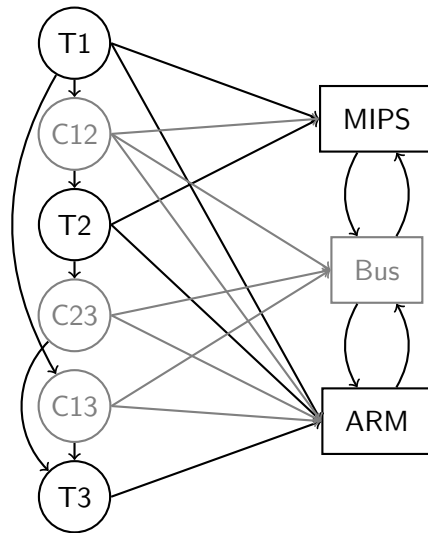


Figure 6: Specification graph.

- (c) (8 points) List all design points (allocation, binding, schedule) and evaluate their maximum CPU utilization ( $f_1$ ) and average bus utilization ( $f_2$ ), based on the following definitions:

$$f_1 = \max(\text{busy}(\text{ARM}), \text{busy}(\text{MIPS}))$$

$\text{busy}(P)$  denotes the long-term average utilization of processor  $P$ .

$$f_2 = \sum_{u \text{ mapped to bus}} \frac{b(u)}{t(\text{bus})}$$

$b(u)$  denotes the long-term average communication request rate in [bit/s] and  $t(\text{bus})$  denotes the bus bandwidth in [bit/s].

Plot the points in Figure 7 and determine the Pareto-optimal solutions.

**Sample solution:**

Table 5: Design points.

Solution	T1	T2	T3	Max. CPU Utilization	Avg. Bus Utilization
# 1	ARM	ARM	ARM	$\max(\frac{(50+10+40)}{100}, 0) = 1.0$	0
# 2	MIPS	ARM	ARM	$\max(\frac{(10+40)}{100}, \frac{(5)}{100}) = 0.5$	$\frac{300}{(5000*0.1)} = 0.6$
#3	ARM	MIPS	ARM	$\max(\frac{(50+40)}{100}, \frac{(30)}{100}) = 0.9$	$\frac{300}{(5000*0.1)} = 0.6$
# 4	MIPS	MIPS	ARM	$\max(\frac{(40)}{100}, \frac{(5+30)}{100}) = 0.4$	$\frac{400}{(5000*0.1)} = 0.8$

There are four design points that need to be considered:

Only Solution # 3 is dominated by another (Solution # 2). All others are in the Pareto Front.

Avg. Bus utilization ( $f_2$ )

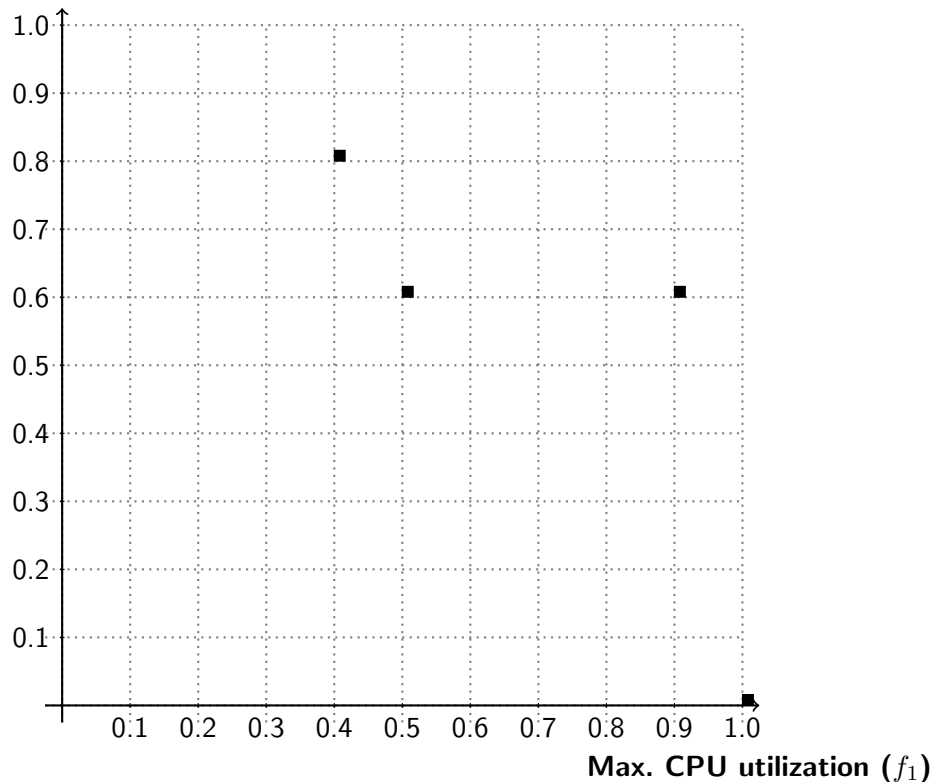


Figure 7: CPU and bus utilization plot.

(d) (4 points) Consider a new Pareto front with two points  $P_a$  and  $P_b$  whose  $(f_1, f_2)$  values

are (0.3,1.0) and (0.5,0), respectively. The multi-objective problem is now transformed to a single objective problem using the following function:

$$\text{cost}(P) = K_1 \times f_1 + K_2 \times f_2$$

This function, which needs to be minimized, assigns a single objective value  $\text{cost}(P)$  to an objective vector  $P$ . Determine the range of  $\frac{K_2}{K_1}$  such that  $P_a$  is the optimal solution.

To determine the ratio of the  $K$  parameters, the costs of points  $P_a$  and  $P_b$  are calculated:

$$\text{cost}(P_a) = 0.3 \cdot K_1 + K_2$$

$$\text{cost}(P_b) = 0.5 \cdot K_1$$

The desired condition is:  $\text{cost}(P_a) < \text{cost}(P_b)$ . Consequently:

$$0.3 \cdot K_1 + K_2 < 0.5 \cdot K_1 \text{ or after simplifying: } \frac{K_2}{K_1} < 0.2.$$

**3.3: Modular Performance Analysis**

(maximal 16 points)

An input stream  $S1$  is processed by CPU, producing an output stream  $S2$ . This stream is in turn sent via BUS. Stream  $S1$  has events arriving with a period of 2ms, jitter of 3ms and zero inter-arrival time. CPU handles its incoming events with a constant execution rate of 1 event/ms. BUS is a TDMA resource with a bandwidth of 1 event/ms, where a slot of 1ms is available in every 2ms cycle.

- (a) (6 points) Draw the upper and lower arrival curves for stream  $S1$ , and the input service curve for CPU, for time intervals up to 8ms. Determine the maximum delay in [ms].

**Sample solution:**

The maximum delay is 2ms – this is the maximal horizontal distance between  $S1$  upper and CPU lower.

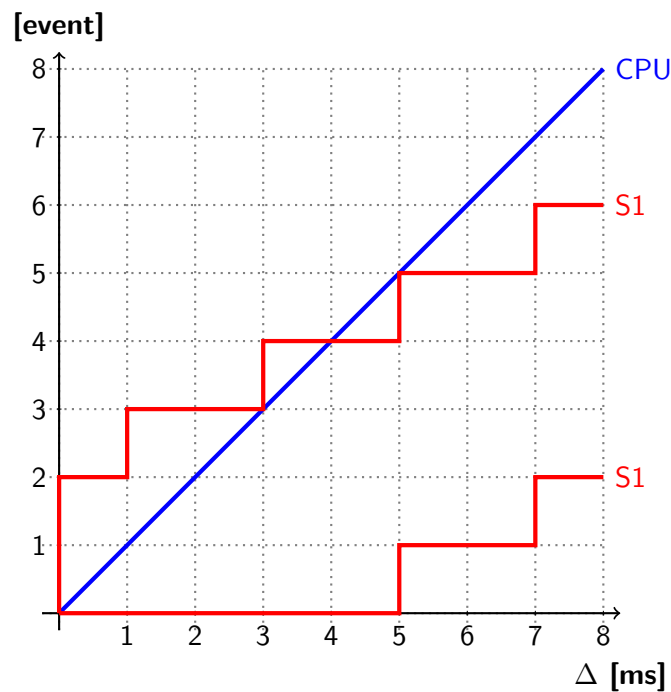


Figure 8: Arrival curves for  $S1$ , and input service curve for CPU.

- (b) (10 points) Draw the upper arrival curve for stream S2, and the upper and lower input service curves for BUS, for time intervals up to 8ms. Determine the maximum buffer capacity needed for S2 in [event], and the maximum delay in [ms].

*Hint:* Assume that the output upper arrival curve  $\alpha^{u'}$  of a component can be computed as:

$$\alpha^{u'} = \alpha^u \circledast \beta^l$$

where  $\alpha^u$  is the upper input arrival curve,  $\beta^l$  is the lower input service curve, and  $f \circledast g$  is the min-plus de-convolution:

$$(f \circledast g)(t) = \sup_{u \geq 0} \{f(t+u) - g(u)\}$$

**Sample solution:**

The maximum capacity of the buffer is 3 events – this is the maximal vertical distance between S2 upper and BUS lower.

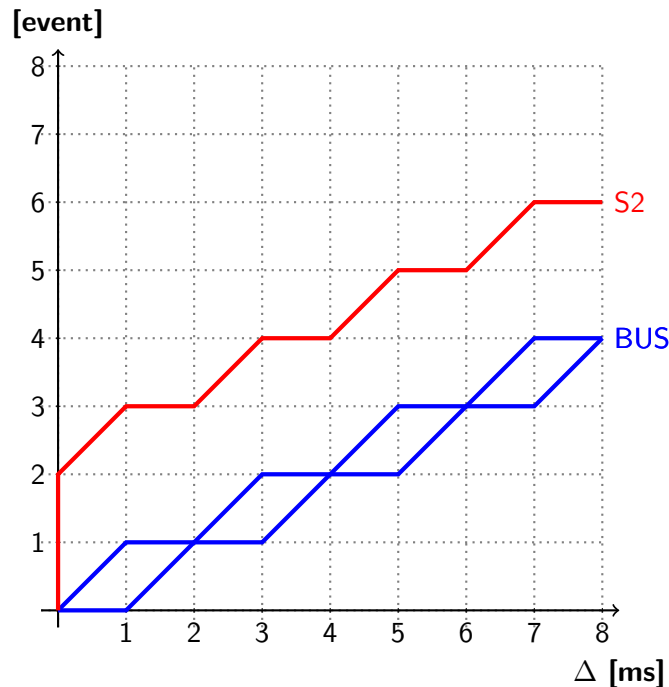


Figure 9: Upper arrival curve for S2, and input service curves for BUS.