Scheduling

Cyclic-executive schedule, Feasible schedule

Exercise class 2

Presenter:
Jürgen Mattheis

In cooperation with:
Pascal Walter

Based on the lecture of:
Marco Zimmerling

November 21, 2022

University of Freiburg, Chair for Embedded Systems

Gliederung

Organisation

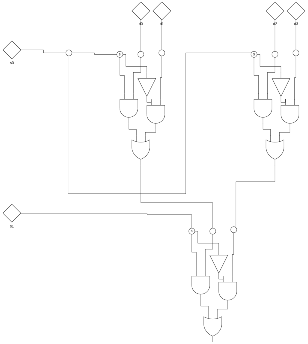
Organisation I

The last exercise class

- this is the english exercise class in Building 101 HS 00-036, the german exercise class is in Building 101 SR 02-016/18
- Correction: in Task 1.2 the k in $M(k)$ is not the number of input lines to select from with the select lines, it's the number of select lines 2^k unlike I said in the last exercise class
  - a multiplexer that selects from 3 lines can be constructed but it doesn't rly make sense for a company to sell such circuits, so we always talk about 2^k-to-1 multiplexers
  - $M(k) = \begin{cases} A_{\text{mux}} = 4 & \text{if } k = 1 \\ 2 \cdot M(k-1) + A_{\text{mux}} & \text{otherwise} \end{cases}.$

Organisation II

The last exercise class



- Questions about Jobs and Projects: `nes-lab.org`
  - click on `Jobs and Theses`

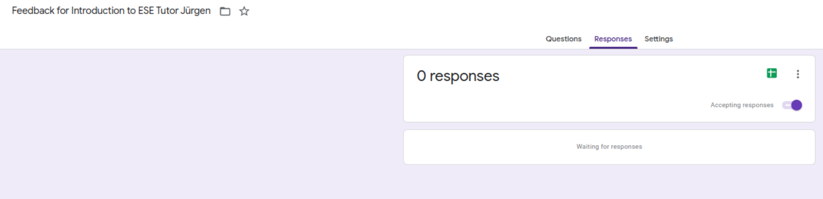
Organisation III

The last exercise class

- Question: Why does the ROM get addressed in bytes?
  - it is correct that in a 32-Bit architecture most of the things are 32-Bit like the registers, the data paths are all 32 bit wide etc. but there's no reason this always has to be the case
  - e.g. in the AT&T Syntax of the X_{86_64} instruction set you can for example write `movl %eax, -4(%ebp)` and this will execute $\%ebp - 4 \leftarrow \%eax$, it will write the content of the `%eax`-register to the byte addressed memory address `%ebp - 4` of the main memory
- feedback for me: `https://forms.gle/f3YN8EFrZ1vsfPoC6`

Organisation IV

The last exercise class



Overview over Scheduling

TT Cyclic Executive Scheduling

Why scheduling?

- In many embedded systems, correct timing is a matter of correctness, not performance.
- Hard real-time systems can be often found in safety-critical applications. If an answer arrives too late within such a system, the consequences can be a catastrophe.
- We want to analyse our systems under a worst case assumption. We need to prove that our system can meet certain deadlines reliably and without statistical arguments.
- Given tasks and their deadlines, we now want to find a suitable arrangement of these periodic tasks such that the system can process them while keeping all their constraints in mind. (Or if not possible, we want to find out why not!)

TT Cyclic Executive Scheduling

Recap: Definitions

- Γ : set of all periodic tasks
- τ_i : one particular periodic task (the i-th)
- $\tau_{i,j}$: the jth instance of task i
- $r_{i,j}$: release time of jth instance of task i
- $d_{i,j}$: absolute deadline of the jth instance of task i
- Φ_i : phase of task i
- D_i : relative deadline of task i

TT Cyclic Executive Scheduling

Recap: Definitions

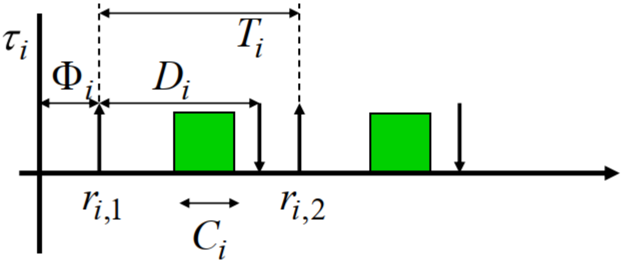


Figure 1: View on a single task

TT Cyclic Executive Scheduling

Recap: Three assumptions

1. The instances of a periodic task are regularly activated at a constant rate. The interval between two consecutive activations is called period. The release times satisfy $r_{i,j} = \Phi_i + (j-1)T_i$
2. All instances have the same worst case execution time C_i (also written as $WCET(i)$)
3. All instances of a periodic task have the same relative deadline D_i. Therefore the absolute deadlines satisfy $d_{i,j} = \Phi_i + (j-1)T_i + D_i$

TT Cyclic Executive Scheduling

Example Schedule

Given $P = 12$ and $f = 4$. Given the table below, find a possible frame assignment

Γ	T_i	Φ_i	D_i	C_i	frame
τ_1	12	2	8	2.8	
τ_2	12	3	9	3	
τ_3	4	0	4	1	

TT Cyclic Executive Scheduling

Example Schedule

Γ	T_i	Φ_i	D_i	C_i	frame
τ_1	12	2	8	2.8	2
τ_2	12	3	9	3	3
τ_3	4	0	4	1	1, 2, 3

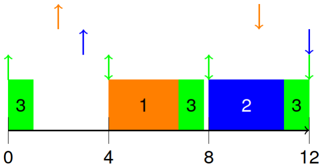


Figure 2: Solution

Task 1

Task 1 I

Feasibility

1. Is the period P a common multiple of all task periods?

   - Yes, as we see...

Task	Period	Deadline	Execution Time	Frames
1	15	9	2	2, 5, 9, 12
2	12	4	3	1, 4, 7, 10, 13
3	10	6	1	1, 3, 6, 8, 11, 13
4	6	6	2	2, 3, 5, 6, 8, 9, 11, 12, 14, 15

Figure 3: A task set and schedule

Task 1 II

Feasibility

2. Is the period P a multiple of the frame f?
   - Yes, $15 \cdot f = P$
3. Is the frame f sufficiently long?
   - Yes. We can see this by drawing the schedule or by adding the execution times per frame...

Task 1 III

Feasibility

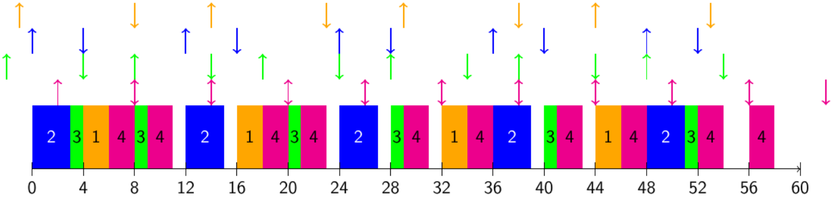


Figure 4: Schedule for Task 1

Task 1 IV

Feasibility

Frame	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
exec. time	$3+1$	$2+2$	$1+2$	3	$2+2$	$1+2$	3	$1+2$
Frame	# 9	# 10	# 11	# 12	# 13	# 14	# 15	
exec. time	$2+2$	3	$1+2$	$2+2$	$3+1$	2	2	

Figure 5: Execution time per frame

4. Determine offsets such that instances start after release time.

- $\Phi_i = \min_{1 \leq j \leq P/T_i} \{(f_{i,j} - 1)\, f - (j-1)\, T_i\}$

Task 1 V

Feasibility

- $$\Phi_1 = \min \begin{cases} (2-1)4-(1-1)15 \\ (5-1)4-(2-1)15 \\ (9-1)4-(3-1)15 \\ (12-1)4-(4-1)15 \end{cases} \min \begin{cases} 4 \\ 1 \\ 2 \\ -1 \end{cases} = -1$$

- $$\Phi_2 = \min \begin{cases} (1-1)4-(1-1)12 \\ (4-1)4-(2-1)12 \\ (7-1)4-(3-1)12 \\ (10-1)4-(4-1)12 \\ (13-1)4-(5-1)12 \end{cases} \min \begin{cases} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{cases} = 0$$

- $$\Phi_3 = \min \begin{cases} (1-1)4-(1-1)10 \\ (3-1)4-(2-1)10 \\ (6-1)4-(3-1)10 \\ (8-1)4-(4-1)10 \\ (11-1)4-(5-1)10 \end{cases} = \min \begin{cases} 0 \\ -2 \\ 0 \\ -2 \\ 0 \end{cases} = -2$$

Task 1 VI

Feasibility

- $$\Phi_4 = \min \begin{cases} (2-1)4-(1-1)6 \\ (3-1)4-(2-1)6 \\ (5-1)4-(3-1)6 \\ (6-1)4-(4-1)6 \\ (8-1)4-(5-1)6 \\ (9-1)4-(6-1)6 \\ (11-1)4-(7-1)6 \\ (12-1)4-(8-1)6 \\ (14-1)4-(9-1)6 \\ (15-1)4-(10-1)6 \end{cases} = \min \begin{cases} 4 \\ 2 \\ 4 \\ 2 \\ 4 \\ 2 \\ 4 \\ 2 \\ 4 \\ 2 \end{cases} = 2$$

5. Are deadlines respected?

- Yes...

Task 1 VII

Feasibility

- $(j-1)T_i + \Phi_i + D_i \geq f_{i,j}f \quad \forall i, 1 \leq j \leq P/T_i$
- $\begin{cases} (1-1)15-1+9 = 8 \geq 8 = 2 \cdot 4 \\ (2-1)15-1+9 = 23 \geq 20 = 5 \cdot 4 \\ (3-1)15-1+9 = 38 \geq 36 = 9 \cdot 4 \\ (4-1)15-1+9 = 53 \geq 48 = 12 \cdot 4 \end{cases}$
- $\begin{cases} (1-1)12+0+4 = 4 \geq 4 = 1 \cdot 4 \\ (2-1)12+0+4 = 16 \geq 16 = 4 \cdot 4 \\ (3-1)12+0+4 = 28 \geq 28 = 7 \cdot 4 \\ (4-1)12+0+4 = 40 \geq 40 = 10 \cdot 4 \\ (5-1)12+0+4 = 52 \geq 52 = 13 \cdot 4 \end{cases}$

Task 1 VIII

Feasibility

- $$\begin{cases} (1-1)10-2+6=4\geq 4=1\cdot 4\\ (2-1)10-2+6=14\geq 12=3\cdot 4\\ (3-1)10-2+6=24\geq 24=6\cdot 4\\ (4-1)10-2+6=34\geq 32=8\cdot 4\\ (5-1)10-2+6=44\geq 44=11\cdot 4\\ (6-1)10-2+6=54\geq 52=13\cdot 4 \end{cases}$$
- $$\begin{cases} (1-1)6+2+6=8\geq 8=2\cdot 4\\ (2-1)6+2+6=14\geq 12=3\cdot 4\\ (3-1)6+2+6=20\geq 20=5\cdot 4\\ (4-1)6+2+6=26\geq 24=6\cdot 4\\ (5-1)6+2+6=32\geq 32=8\cdot 4\\ (6-1)6+2+6=38\geq 36=9\cdot 4\\ (7-1)6+2+6=44\geq 44=11\cdot 4\\ (8-1)6+2+6=50\geq 48=12\cdot 4\\ (9-1)6+2+6=56\geq 56=14\cdot 4 \end{cases}$$

Task 1 IX

Feasibility

Task 2

Task 2 I

Manual Scheduling

- We see from the table that the period P is 30, and we can use 3 as the frame f. Since this task set is a small one, we can derive a feasible schedule graphically...
  - $P = lcm(15, 10, 6) = 30$
  - for f one does have the constraints: "Is the period P a multiple of the frame f", $f \leq T_i, \forall$ tasks τ_i, $f \geq C_i, \forall$ tasks τ_i and $2f - \gcd(T_i, f) \leq D_i \forall$ tasks $\tau_i \rightarrow 3$
  - Task 1: $2f - gcd(15, f) \leq 3$ and $f \in \{3, 5, 6\}$
    - $f = 3$: $6 - 3 \leq 3$ ✓
    - $f = 5$: $10 - 5 \leq 3$

Task 2 II

Manual Scheduling

- $f = 6$: $12 - 3 \leq 3$

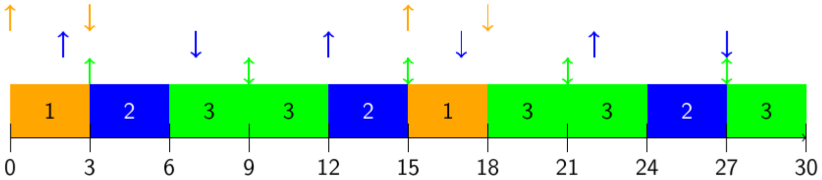


Figure 6: Schedule for Task 2

Task 3

Task 3 I

Bonus Practice

- We see from the table that the period P is 30, and we can use 3 as the frame f. Since this task set is a small one, we can derive a feasible schedule graphically...
  - Task 1: $2f - gcd(15, f) \leq 3$ and $f \in \{2, 3, 5, 6\}$
    - $f = 2$: $4 - 1 \leq 3$ ✓
    - $f = 3$: $6 - 3 \leq 3$ ✓
    - $f = 5$: $10 - 5 \leq 3$
    - $f = 6$: $12 - 3 \leq 3$
  - Task 2: $2f - gcd(10, f) \leq 5$ and $f \in \{2, 3\}$

Task 3 II

Bonus Practice

  - $f = 2$: $4 - 2 \leq 5$ ✓
  - $f = 3$: $6 - 1 \leq 5$ ✓
- Task 3 and 4: $2f - gcd(6, f) \leq 5$ and $f \in \{2, 3\}$
  - $f = 2$: $4 - 2 \leq 5$ ✓
  - $f = 3$: $6 - 3 \leq 5$ ✓

Task 3 III

Bonus Practice

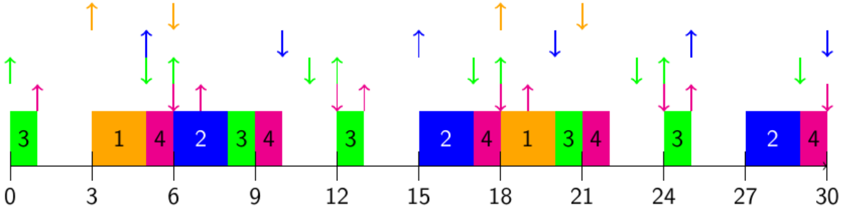


Figure 7: Schedule for Task 3