Prof. Dr. Florian Künzner
Technical University of Applied Sciences Rosenheim, Computer Science

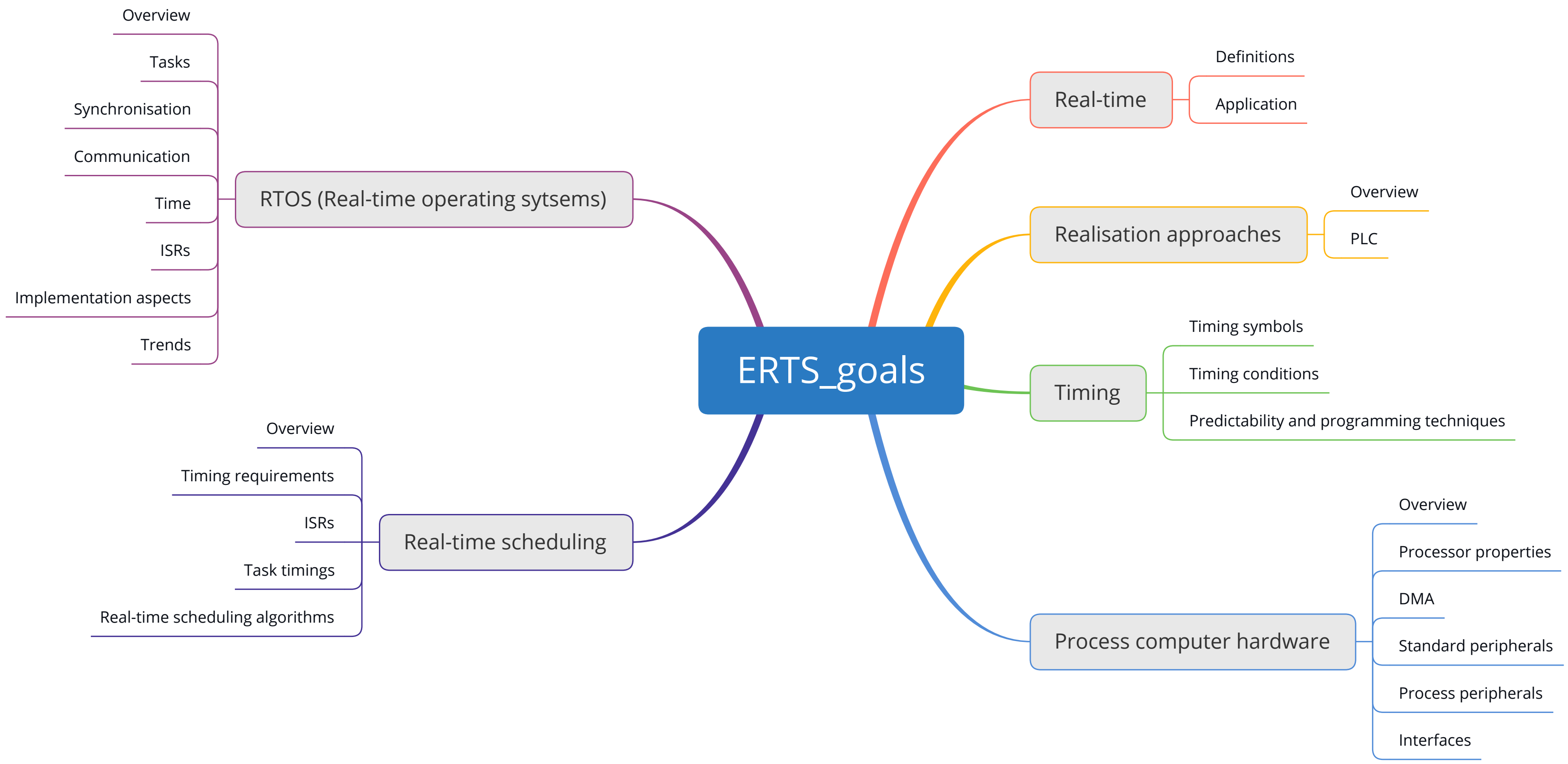
ERTS - Embedded real-time systems

ERTS 6 – Process computer HW 3

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal

ERTS::Process computer HW 3

- Overview
- DMA interference
- DMA system load examples
- DMA system load definitions
- DMA Exercises

CAMPUS Rosenheim
Computer Science

Overview

How to transfer data between devices and the RAM?

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Overview

I/O programming methods*:

- **Programmed I/O** (busy wait or polling)
  - – CON: CPU has to check if device is ready (for every word transfer)
  - – CON: CPU has to perform the data transfer
- **Interrupt driven I/O**
  - + PRO: CPU is informed when device is ready (for every word transfer)
  - – CON: CPU has to perform the data transfer
- **Direct memory access (DMA)**
  - + PRO: BUS devices directly transfer data
  - + PRO: CPU is informed after everything has been finished

*recap from RA lecture

CAMPUS Rosenheim
Computer Science

DMA recap

Recap from **CA** lecture.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Interrupt driven I/O

System load example for
interrupt driven I/O

(a) Interrupt driven I/O (single word transfer)

Given:

- I/O transfer rate: $R_{TIO} = 64000$ words/s
- Bus width: 1 word = 16 bit
- To transfer: 1 word, 10 instructions (1 inst. $\approx$ 1 cycle $\rightarrow$ 10 cycles) are required for the ISR
- The processor runs with 1 MHz

Question:

- What is the resulting processor utilisation U_P (also known as system load)?
- Transfer rate R_T in kB/s?

Proposed solution:

- $T_E = 10 \times \frac{1}{1\ \text{MHz}} = 10 \times \frac{1}{1 \times 10^6} = 0.00001\ \text{s} \Rightarrow 10\ \mu\text{s}$
- $T_P = \frac{1}{64000} = 0{,}000015625\ \text{s} \Rightarrow 15.625\ \mu\text{s}$
- $U_P = \frac{T_E}{T_P} = \frac{10\ \mu\text{s}}{15.625\ \mu\text{s}} = 0.64 \Rightarrow 64\ \%$
- Transfer rate: $R_T = 64000 \times \frac{16}{8} = 128000\ \text{B} \Rightarrow 128\ \text{kB/s}$

CAMPUS Rosenheim
Computer Science

Interrupt driven I/O

Can **DMA** improve the situation?

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

DMA interference

Cycle stealing method **(while the DMA device is active):**

- Both, the I/O transfer and the CPU runs in parallel
- The CPU can't use the memory
- If the CPU needs the memory (read/write), the CPU has to wait until the DMA cycle is completed

Problem:

- No way of predicting how many times the CPU will have to wait for the DMA cycle completion

$\Rightarrow$ Response time of running task cannot be precisely determined

Another method: ***time-slice method*** [SR88]

- Each memory cycle is split into two adjacent time slots
- One for the CPU, the other for the DMA device

+ Pro: More predictable, due to no or less CPU waiting conditions

− Con: More expensive than the cycle stealing

source: [2, Buttazzo, p. 14]
[SR88] J. Stankovic and K. Ramamritham, editors. Tutorial on Hard Real-Time Systems. IEEE Computer Society Press, 1988.

CAMPUS Rosenheim
Computer Science

DMA: single word transfer vs burst mode

For the **DMA**, a distinction is made between **single word transfers** and a **burst mode** with $b_{max} = 1, \ldots, \infty$.

Hint: **Large** b_{max} are **problematic** for real-time systems, because it can reduce the reactivity and introduce some **non-determinism** in **predicting** the **WCET**.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

DMA system load definitions

What is the influence of the **DMA**: Load or bus utilisation and the increased, resulting runtime?

CAMPUS Rosenheim
Computer Science

(a) DMA (single word transfer)

Resulting system load or bus utilisation through DMA:

β or U_{BUS}	$\rightarrow$	Load or bus utilisation $\beta = n \times T_{\text{DMA}_c}$
n	$\rightarrow$	Number of words per second
T_{DMA_c}	$\rightarrow$	Time for one DMA cycle

Details:

- Typical values for $T_{\text{DMA}_c} = 500, \ldots, 100$ ns for slower systems (VMEBus)
- For normal PCs, T_{DMA_c} is faster

Resulting runtime:

- There is a maximum approximated increase of the runtime by the factor: $(1+\beta)$
- Resulting runtime $\approx$ runtime for a calculation or a task $\times (1+\beta)$
- Attention: This is only valid for small n; the exact factor is $\frac{1}{1-\beta}$

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

(b) DMA (burst mode, fixed block size)

Resulting system load or bus utilisation through DMA:

β or U_{BUS}	$\rightarrow$	Load or bus utilisation $\beta = n_b \times T_{\text{DMA}_b}$
n_b	$\rightarrow$	Number of burst blocks per second
T_{DMA_b}	$\rightarrow$	Time to transfer one burst block: $T_{\text{DMA}_b} = T_{\text{latency}} + b_{\text{max}} \times T_{\text{DMA}_c}$
T_{latency}	$\rightarrow$	Waiting time after a burst block

Resulting runtime:

- There is a maximum approximated increase of the runtime by the factor: $\frac{1}{1-\beta}$
- Resulting runtime $\approx$ runtime for a calculation or a task $\times \frac{1}{1-\beta}$

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

DMA system load examples

Detailed analysis:

- DMA (single word transfer, without cache, for theoretical purposes only)
- DMA (burst mode, normal load)
- DMA (extreme/high load)

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

(b) DMA (single word transfer)

Details: without cache, for theoretical purposes only

Given:

- To transfer: $R_{TIO} = 64000$ words/s
- Bus width: 1 word = 16 bit
- A/D converter (ADC) generates 1 word every 15.625 μs

$\Rightarrow$ Rate: $R_{ADC} = \frac{1}{15.625\ \mu s} = 64000$ samples/s

- Bus cycle time: 150 ns

Question:

- What is the resulting load or bus utilisation β through the cycle stealing?
- What is the maximum transfer rate $R_{T_{\text{BUS}}}$ of the bus?

Proposed solution:

- Load: $\beta = 64000 \times 150\ \text{ns} = 0.0096 \approx 0.01 \Rightarrow 1\ \%$
- Maximum transfer rate of bus:
  $R_{T_{\text{BUS}}} = \frac{1}{150\ \text{ns}} \times \frac{16}{8} = \frac{1}{150 \times 10^{-9}} \times 2 = 13333333.33\ \text{B} \Rightarrow \approx 13.33\ \text{MB/s}$

But: Modern systems are more complex!

CAMPUS Rosenheim
Computer Science

(c) DMA (burst mode, normal load)

Given:

- To transfer: 64000 words/s, each word has again 16 bit.
- A/D converter (ADC) generates 1 word every 15.625 μs; $R_{ADC} = \frac{1}{15.625\ \mu s} = 64000$ samples
- Bus width: 1 word = 64 `bit`
- Transfer time: 2.5 ns for one word; $T_{\text{latency}} = 50$ ns (after a block transfer)
- $b_{\text{max}} = 8$; all memory accesses and transfers are 64 byte blocks

Question:

- What is the resulting load or bus utilisation β through the cycle stealing?

Proposed solution:

- Number of burst blocks per second: $n_b = \frac{64000}{8} = 8000$
- Time to transfer one burst block: $T_{\text{DMA}_b} = 50\text{ ns} + 8 \times 2.5\text{ ns} = 70\text{ ns}$
- Load: $\beta = 8000 \times 70\text{ ns} = 0.00056 \Rightarrow 0.056\ \%$

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

(d) DMA (burst mode, high load)

Given:

- To transfer: 1000×64000 words/s $= 64{,}000{,}000$ words/s, each word has again 16 bit.
- $R_{ADC} = 1000 \times 64000$ samples
- Bus width: 1 word $= 64$ `bit`
- Transfer time: 2.5 ns for one word; $T_{\text{latency}} = 50$ ns (after a block transfer)
- $b_{\text{max}} = 8$; all memory accesses and transfers are 64 byte blocks

Question:

- What is the resulting load or bus utilisation β through the cycle stealing?

Proposed solution:

- Number of burst blocks per second: $n_b = \frac{1000 \times 64000}{8} = 80{,}000{,}000$
- Load: $\beta = 80{,}000{,}000 \times 70 \text{ ns} = 0.56 \Rightarrow 56~\%$

CAMPUS Rosenheim
Computer Science

Exercise 1: Processor utilisation
Evaluation of an interrupt driven I/O system

Given:

- I/O transfer rate: $R_{TIO} = 40000$ words/s
- Bus width: 1 word = 32 bit
- ISR takes $T_E = 20\ \mu s$

Question:

- Is it possible to safely handle the load of the I/O? *Hint: You may calculate T_P and U_P to answer that.*
- Will interrupts be lost?
- Also consider the influence of the operating system!

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

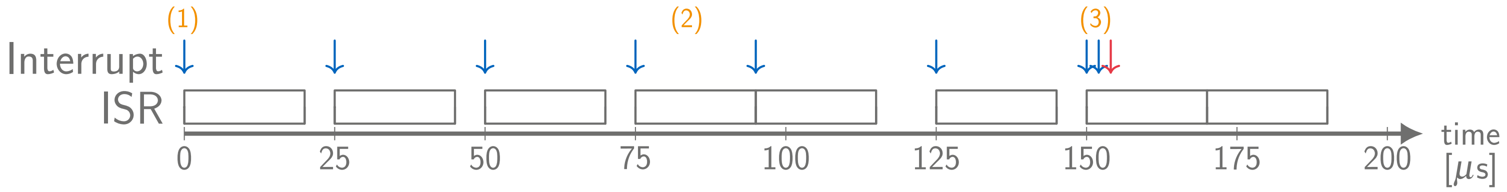
Exercise 1: Processor utilisation

Evaluation of an interrupt driven I/O system

Proposed solution:

- $T_P = \frac{1}{40000} = 0.000025 \text{ s} \Rightarrow 25\ \mu\text{s}$
- $U_P = \frac{T_E}{T_P} = \frac{20\ \mu\text{s}}{25\ \mu\text{s}} = 0.8 \Rightarrow 80\ \%$

Interrupt period analysis:



1. If the interrupts are equally distributed (small variance/jitter), then this might be possible
2. Here, it's very close with no safety buffer
3. Here, interrupt signals are lost

Evaluation:

- On a hard real-time system (and safety), usually, (1), (2), and (3) are not acceptable
- For a soft or firm real-time system, (1) may be acceptable (but usually not)
- If an OS is used and if it disables the interrupt for some time, this might further complicate the situation

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise 2: DMA system load

Evaluation of an I/O system using DMA

Given:

- I/O transfer: 10 MiB within 1 s
- Bus width: 1 word = 32 bit
- Block length = 64 bytes

Question:

- Is it possible to safely handle the load of the I/O? *Hint: You may calculate the load β to answer that.*
- Will interrupts be lost?
- Also consider the influence of the operating system!

CAMPUS Rosenheim
Computer Science

Exercise 2: DMA system load

Evaluation of an I/O system using DMA

Proposed solution:

- 1 word: $\frac{32}{8} = 4$ byte; $b_{max} = \frac{64}{4} = 16$
- $n_b = \frac{10\ \text{MiB}}{64\ \texttt{bytes}} = \frac{10485760}{64} = 163840$ burst blocks per second
- $T_{\text{DMA}_b} = T_{\text{latency}} + b_{\text{max}} \times T_{\text{DMA}_c} = 50\ \text{ns} + 16 \times 2.5\ \text{ns} = 90\ \text{ns}$
- $\beta = n_b \times T_{\text{DMA}_b} = 163840 * 90\ \text{ns} = 0.0147456 \approx 0.015 \Rightarrow 1.5\ \%$
- Maximum approximated increase of the runtime by the factor: $\frac{1}{1-0.015} \approx 1.015$

Evaluation:

- The increase of the runtime by the factor of ≈ 1.015 is low and can usually be accepted for soft, firm, and hard real-time systems
- Also, an RTOS with a limited time where the interrupts are disabled should be acceptable

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Exercise 1+2: System load

Evaluation of an combined interrupt driven I/O and DMA system

Question:

- Can the load of example 1 and 2 be combined?

Proposed solution:

- Estimation: $U \approx U_P + \beta = 0.8 + 0.015 = 0.815$
- $T_E = 20\ \mu s * 1.015 \approx 20.3\ \mu s$
- $U = \frac{20.3\ \mu s}{25\ \mu s} = 0.812 \Rightarrow 81.2\ \%$

Evaluation:

- The additional load introduced by DMA plays practically no role
- But, as stated in example 1, the load of $\gtrsim 80\%$ is usually not acceptable

CAMPUS Rosenheim
Computer Science

Summary and outlook

Summary

- Overview
- DMA interference
- DMA system load examples
- DMA system load definitions
- DMA Exercises

Outlook

- Scheduling theory