Prof. Dr. Florian Künzner

Technical University of Applied Sciences Rosenheim, Computer Science

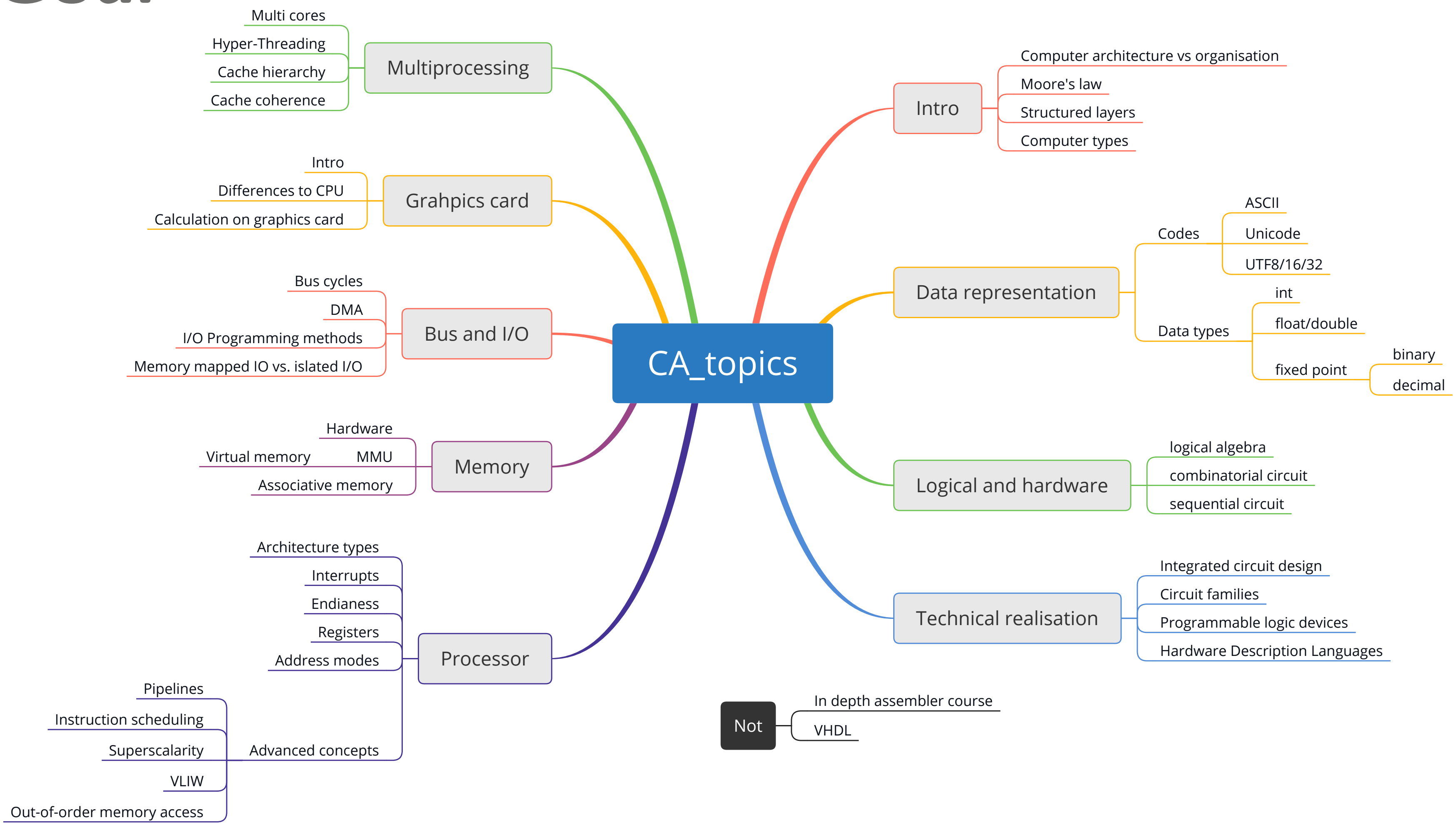
CA 4 – Technical realisation

The lecture is based on the work and the documents of Prof. Dr. Theodor Tempelmeier

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal



- CA_topics
  - Intro
    - Computer architecture vs organisation
    - Moore's law
    - Structured layers
    - Computer types
  - Data representation
    - Codes
      - ASCII
      - Unicode
      - UTF8/16/32
    - Data types
      - int
      - float/double
      - fixed point
        - binary
        - decimal
  - Logical and hardware
    - logical algebra
    - combinatorial circuit
    - sequential circuit
  - Technical realisation
    - Integrated circuit design
    - Circuit families
    - Programmable logic devices
    - Hardware Description Languages
  - Processor
    - Architecture types
    - Interrupts
    - Endianess
    - Registers
    - Address modes
    - Advanced concepts
      - Pipelines
      - Instruction scheduling
      - Superscalarity
      - VLIW
      - Out-of-order memory access
  - Memory
    - Hardware
    - Virtual memory
      - MMU
    - Associative memory
  - Bus and I/O
    - Bus cycles
    - DMA
    - I/O Programming methods
    - Memory mapped IO vs. islated I/O
  - Grahpics card
    - Intro
    - Differences to CPU
    - Calculation on graphics card
  - Multiprocessing
    - Multi cores
    - Hyper-Threading
    - Cache hierarchy
    - Cache coherence
- Not
  - In depth assembler course
  - VHDL

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Goal

CA::Technical realisation

- Development of integrated circuits
- Circuit families
- Programmable Logic Devices

CAMPUS Rosenheim
Computer Science

Logical functionality

...and it’s technical realisation

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Development of integrated circuits (ICs)

CAMPUS Rosenheim
Computer Science

Development of integrated circuits (ICs)

Description with „(HDL) hardware description language“ (simulation)

```
y = a * b
```

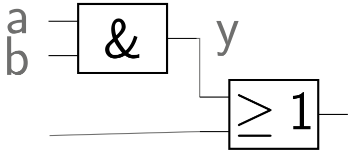
CAMPUS Rosenheim
Computer Science

Development of integrated circuits (ICs)

Description with „(HDL) hardware description language“ (simulation)

```
y = a * b
```

Logic (CAD, simulation)



CAMPUS Rosenheim
Computer Science

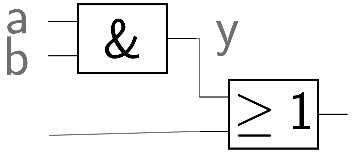
Technische Hochschule Rosenheim
Technical University of Applied Sciences

Development of integrated circuits (ICs)

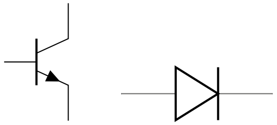
Description with „(HDL) hardware description language“ (simulation)

```
y = a * b
```

Logic (CAD, simulation)



Technical – principle (CAD, simulation)



CAMPUS Rosenheim
Computer Science

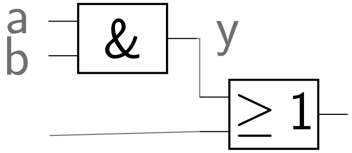
Technische Hochschule Rosenheim
Technical University of Applied Sciences

Development of integrated circuits (ICs)

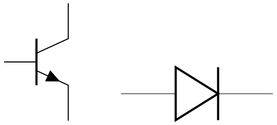
Description with „(HDL) hardware description language“ (simulation)

```
y = a * b
```

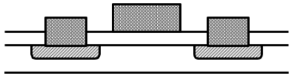
Logic (CAD, simulation)



Technical – principle (CAD, simulation)



Technical – layout and layers on a chip (CAD, simulation)



Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Circuit families

Bipolar

Transistor transistor logic (TTL)

Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)

Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

+ fast
- high power dissipation (Verlustleistung)
- low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- not that fast
+ low power dissipation
+ high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Circuit families

Bipolar

Transistor transistor logic (TTL)
Emitter coupled logic (ECL)

- + fast
- – high power dissipation (Verlustleistung)
- – low integration density

Unipolar

Metal oxide semiconductor (MOS)
Field effect transistor (FET)

- – not that fast
- + low power dissipation
- + high integration density

Use in microprocessors and memory devices.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Programmable logic devices

Types of IC's (integrated circuits):

1. Standard IC (the whole logic is predefined)
2. Full custom - IC

   Chip with customer logic (may be expensive and time consuming until production finished)
3. Gate arrays

   - Chip with a lot of logic elements (1. production step is the same for everyone)

   - Connection of the gates according to customer specifications (2. production step with mask according to customer specifications)
4. Field programmable logic modules
   (PLA, FPLA, PAL, EPLD, EEPLD, **GAL**, ..., **FPGA**)
   - "On site" with PC and programming device or even in soldered in state "programmable" - Programming means put user-specific logic into the IC.

Terminology:

The ICs under the numbers 2 and 3 (sometimes also 4) are called ASICs.
ASIC = application specific IC

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Programmable logic devices

Types of IC's (integrated circuits):

1. Standard IC (the whole logic is predefined)
2. Full custom - IC
   Chip with customer logic (may be expensive and time consuming until production finished)
3. Gate arrays
   - Chip with a lot of logic elements (1. production step is the same for everyone)
   - Connection of the gates according to customer specifications (2. production step with mask according to customer specifications)
4. Field programmable logic modules
   (PLA, FPLA, PAL, EPLD, EEPLD, **GAL**, ..., **FPGA**)
   - "On site" with PC and programming device or even in soldered in state "programmable" - Programming means put user-specific logic into the IC.

Terminology:

The ICs under the numbers 2 and 3 (sometimes also 4) are called ASICs.
ASIC = application specific IC

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Programmable logic devices

Types of IC's (integrated circuits):

1. Standard IC (the whole logic is predefined)
2. Full custom - IC

   Chip with customer logic (may be expensive and time consuming until production finished)
3. Gate arrays

   - Chip with a lot of logic elements (1. production step is the same for everyone)

   - Connection of the gates according to customer specifications (2. production step with mask according to customer specifications)
4. Field programmable logic modules

   (PLA, FPLA, PAL, EPLD, EEPLD, **GAL**, ..., **FPGA**)

   - "On site" with PC and programming device or even in soldered in state "programmable" - Programming means put user-specific logic into the IC.

Terminology:

The ICs under the numbers 2 and 3 (sometimes also 4) are called ASICs.
ASIC = application specific IC

Programmable logic devices

Types of IC's (integrated circuits):

1. Standard IC (the whole logic is predefined)
2. Full custom - IC

   Chip with customer logic (may be expensive and time consuming until production finished)
3. Gate arrays

   - Chip with a lot of logic elements (1. production step is the same for everyone)

   - Connection of the gates according to customer specifications (2. production step with mask according to customer specifications)
4. Field programmable logic modules

   (PLA, FPLA, PAL, EPLD, EEPLD, **GAL**, ..., **FPGA**)

   - "On site" with PC and programming device or even in soldered in state "programmable" - Programming means put user-specific logic into the IC.

Terminology:

The ICs under the numbers 2 and 3 (sometimes also 4) are called ASICs.

ASIC = application specific IC

Programmable logic devices

Types of IC's (integrated circuits):

1. Standard IC (the whole logic is predefined)
2. Full custom - IC

   Chip with customer logic (may be expensive and time consuming until production finished)
3. Gate arrays

   - Chip with a lot of logic elements (1. production step is the same for everyone)

   - Connection of the gates according to customer specifications (2. production step with mask according to customer specifications)
4. Field programmable logic modules

   (PLA, FPLA, PAL, EPLD, EEPLD, **GAL**, ..., **FPGA**)

   - "On site" with PC and programming device or even in soldered in state "programmable" - Programming means put user-specific logic into the IC.

Terminology:

The ICs under the numbers 2 and 3 (sometimes also 4) are called ASICs.

ASIC = application specific IC

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Programmable logic devices

Types of IC's (integrated circuits):

1. Standard IC (the whole logic is predefined)
2. Full custom - IC

   Chip with customer logic (may be expensive and time consuming until production finished)
3. Gate arrays

   - Chip with a lot of logic elements (1. production step is the same for everyone)

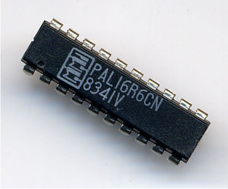
   - Connection of the gates according to customer specifications (2. production step with mask according to customer specifications)
4. Field programmable logic modules
   (PLA, FPLA, PAL, EPLD, EEPLD, **GAL**, ..., **FPGA**)
   - "On site" with PC and programming device or even in soldered in state "programmable" - Programming means put user-specific logic into the IC.

Terminology:

The ICs under the numbers 2 and 3 (sometimes also 4) are called ASICs.

ASIC = application specific IC

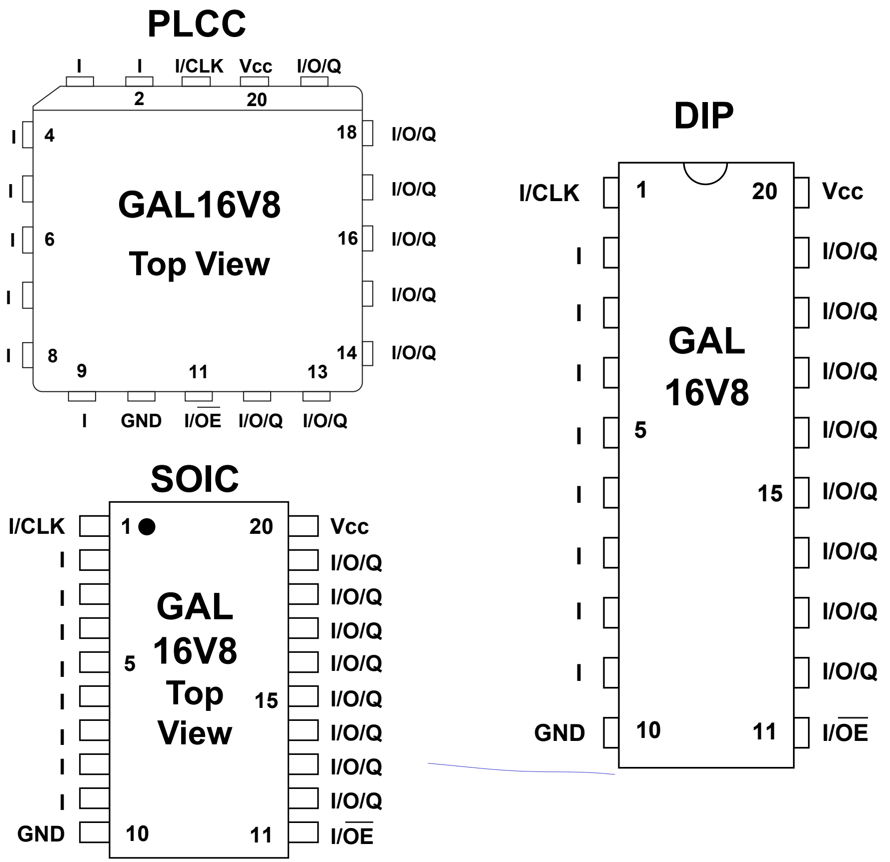
Example: Field programmable logic modules

PAL	Programmable array logic	
GAL	Generic array logic	
FPGA	Field programmable gate array	

[Image sources: wikipedia.org]

Example: GAL (generic array logic)

- Type: GAL16V8
- Company: Lattice
- Programmable AND array
- 20 year data retention



CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Questions?

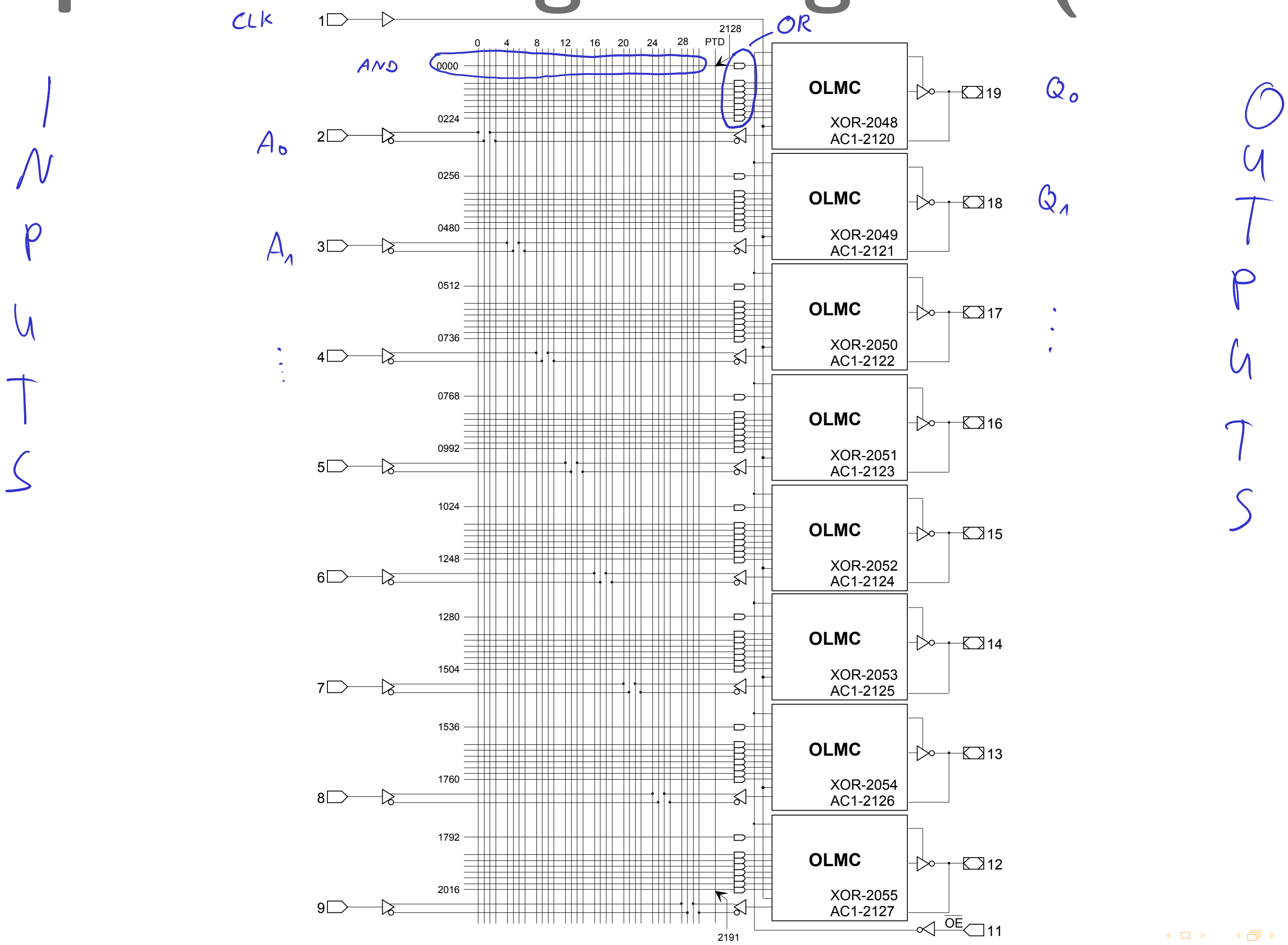
All right? ⇒ ✓

Question? ⇒ ✋ and use **chat**

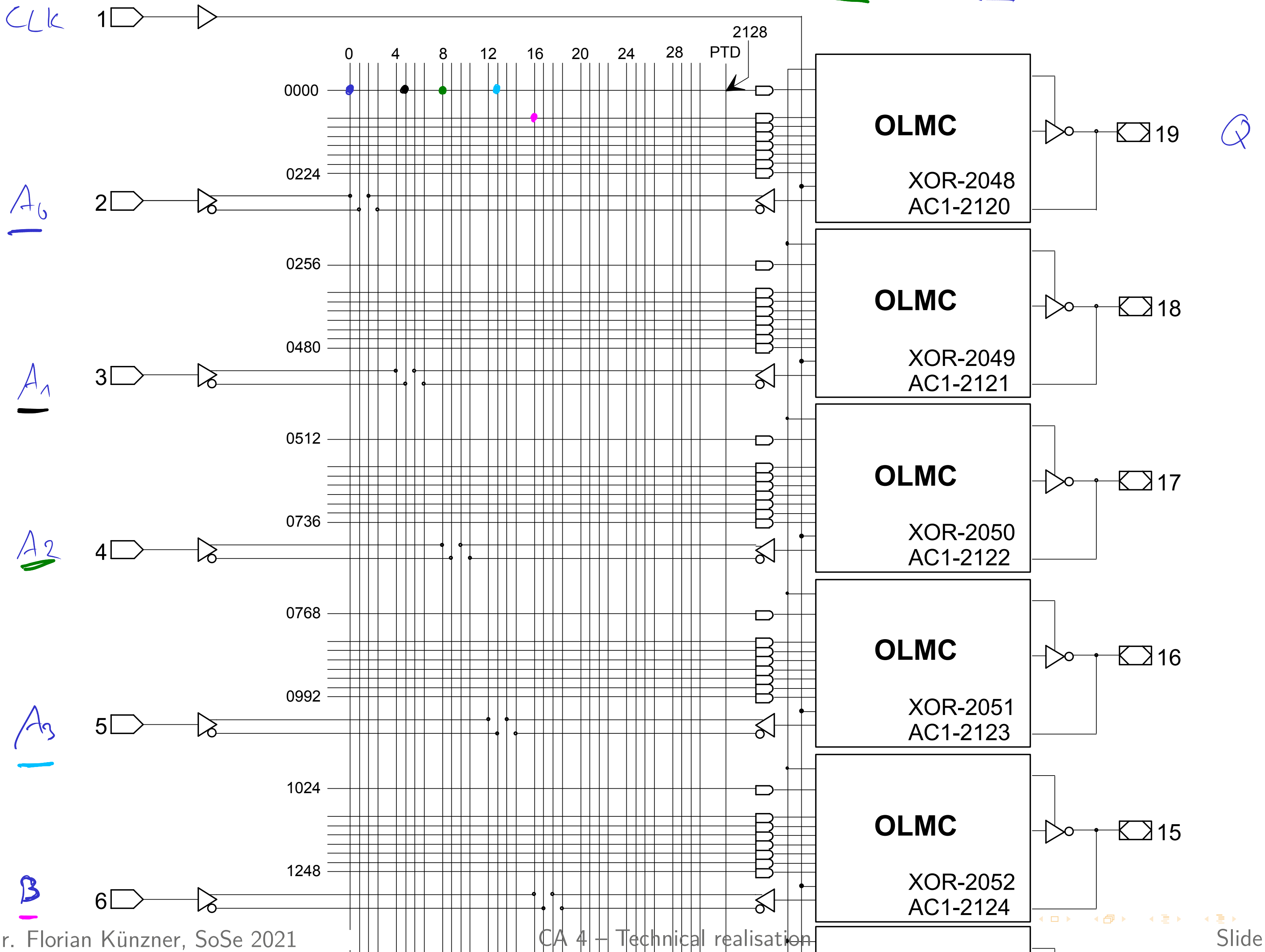
or

speak *after* I ask you to

Example: GAL logic diagram (GAL16V8)



Example: GAL programming: $Q = \bar{A}_3 \cdot A_2 \cdot \bar{A}_1 \cdot A_0 + B$



Questions?

All right? ⇒ ✓

Question? ⇒ ✋ and use **chat**

or

speak *after* I ask you to

Tango-PLD

A language to describe the logic.

```
Logic(in A0, A1, A2, A3, B; out Q){
    Q = !A3 & A2 & !A1 & A0 | B;
    putpart("g16v8", "Logic",
            -,A0,A1,A2,A3,B,-,-,-,GND,
            -,-,-,-,-,-,-,-,Q,VCC);
}
```

- The Tango-PLD compiler automatically creates the contact pattern (JEDEC file, zeros and ones)
- JEDEC file (Joint Electron Device Engineering Council):
  File format between data preperation system and PLD programmer
- Tango-PLD is outdated, because it's too simple for modern FPGAs.

[PLD file formats]

Tango-PLD

A language to describe the logic.

```
Logic(in A0, A1, A2, A3, B; out Q){
    Q = !A3 & A2 & !A1 & A0 | B;
    putpart("g16v8", "Logic",
            -,A0,A1,A2,A3,B,-,-,-,GND,
            -,-,-,-,-,-,-,-,Q,VCC);
}
```

- The Tango-PLD compiler automatically creates the contact pattern (JEDEC file, zeros and ones)
- JEDEC file (Joint Electron Device Engineering Council):
  File format between data preperation system and PLD programmer
- Tango-PLD is outdated, because it's too simple for modern FPGAs.

[PLD file formats]

Tango-PLD

A language to describe the logic.

```
Logic(in A0, A1, A2, A3, B; out Q){
    Q = !A3 & A2 & !A1 & A0 | B;
    putpart("g16v8", "Logic",
            -,A0,A1,A2,A3,B,-,-,-,GND,
            -,-,-,-,-,-,-,-,Q,VCC);
}
```

- The Tango-PLD compiler automatically creates the contact pattern (JEDEC file, zeros and ones)
- JEDEC file (Joint Electron Device Engineering Council):
  File format between data preperation system and PLD programmer
- Tango-PLD is outdated, because it's too simple for modern FPGAs.

[PLD file formats]

Tango-PLD

A language to describe the logic.

```
Logic(in A0, A1, A2, A3, B; out Q){
    Q = !A3 & A2 & !A1 & A0 | B;
    putpart("g16v8", "Logic",
            -,A0,A1,A2,A3,B,-,-,-,GND,
            -,-,-,-,-,-,-,-,Q,VCC);
}
```

- The Tango-PLD compiler automatically creates the contact pattern (JEDEC file, zeros and ones)
- JEDEC file (Joint Electron Device Engineering Council):
  File format between data preperation system and PLD programmer
- Tango-PLD is outdated, because it's too simple for modern FPGAs.

[PLD file formats]

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Hardware description languages

VHDL – Very high speed integrated circuit hardware description language

⇒ **Programming of FPGAs!**

- A precise, formal description of an electronic circuit
- Allows automated analysis and simulation of an electronic circuit
- HDL != Programming language
  (HDL explicitly consider the notion of time and is therefore more complex)
- The syntax of VHDL is very similar to the programming language Ada.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Hardware description languages

VHDL – Very high speed integrated circuit hardware description language

⇒ **Programming of FPGAs!**

- A precise, formal description of an electronic circuit
- Allows automated analysis and simulation of an electronic circuit
- HDL ! = Programming language
  (HDL explicitly consider the notion of time and is therefore more complex)
- The syntax of VHDL is very similar to the programming language Ada.

Hardware description languages

VHDL – Very high speed integrated circuit hardware description language

⇒ **Programming of FPGAs!**

- A precise, formal description of an electronic circuit
- Allows automated analysis and simulation of an electronic circuit
- HDL ! = Programming language
  (HDL explicitly consider the notion of time and is therefore more complex)
- The syntax of VHDL is very similar to the programming language Ada.

Hardware description languages

VHDL – Very high speed integrated circuit hardware description language

⇒ **Programming of FPGAs!**

- A precise, formal description of an electronic circuit
- Allows automated analysis and simulation of an electronic circuit
- HDL ! = Programming language
  (HDL explicitly consider the notion of time and is therefore more complex)
- The syntax of VHDL is very similar to the programming language Ada.

Hardware description languages

VHDL – Very high speed integrated circuit hardware description language

⇒ **Programming of FPGAs!**

- A precise, formal description of an electronic circuit
- Allows automated analysis and simulation of an electronic circuit
- HDL ! = Programming language
  (HDL explicitly consider the notion of time and is therefore more complex)
- The syntax of VHDL is very similar to the programming language Ada.

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Questions?

All right? ⇒ ✓

Question? ⇒ ✋ and use **chat**

or

speak *after* I ask you to

CAMPUS Rosenheim
Computer Science

Technische Hochschule Rosenheim
Technical University of Applied Sciences

Summary and outlook

Summary

- Development of integrated circuits
- Circuit families
- Programmable Logic Devices

Outlook

- Processor architecture

CAMPUS Rosenheim
Computer Science

Summary and outlook

Summary

- Development of integrated circuits
- Circuit families
- Programmable Logic Devices

Outlook

- Processor architecture