

# Digitaltechnik

## Wintersemester 2017/2018

### 14. Vorlesung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT





1. Einleitung
2. Lehrevaluation der FB 20 Fachschaft
3. Field Programmable Gate Arrays (FPGAs)
4. Abschluss Digitaltechnik
5. Zusammenfassung

# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

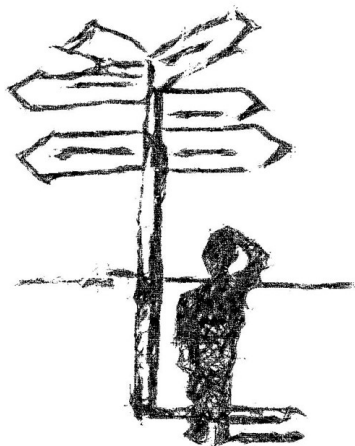
1. Einleitung

2. Lehrevaluation der FB 20 Fachschaft

3. Field Programmable Gate Arrays (FPGAs)

4. Abschluss Digitaltechnik

5. Zusammenfassung





- ▶ Ende von Übungen und Testaten in KW 5
- ▶ Testatsergebnisse (485 Anmeldungen für Studienleistung):

bestandene Testate	1	2	3	4
Anzahl Studierende	105	333	9	1

- ▶ 104 Teilnehmer haben noch Chance auf Klausurzulassung
- ▶ davon aber nur 31 für Testat in KW 5 angemeldet

⇒ letzte Chance bis 02.02.18 nutzen

# Rückblick auf die letzte Vorlesung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Weitere arithmetische Grundsaltungen
  - ▶ Schnelle Additionen
  - ▶ kombinatorische und sequentielle Multiplikation
- ▶ Ausblick Rechnerorganisation
  - ▶ Von-Neumann- und Harvard-Architektur
  - ▶ Speicher
  - ▶ Arithmetisch-Logische Einheit
  - ▶ Steuerwerk



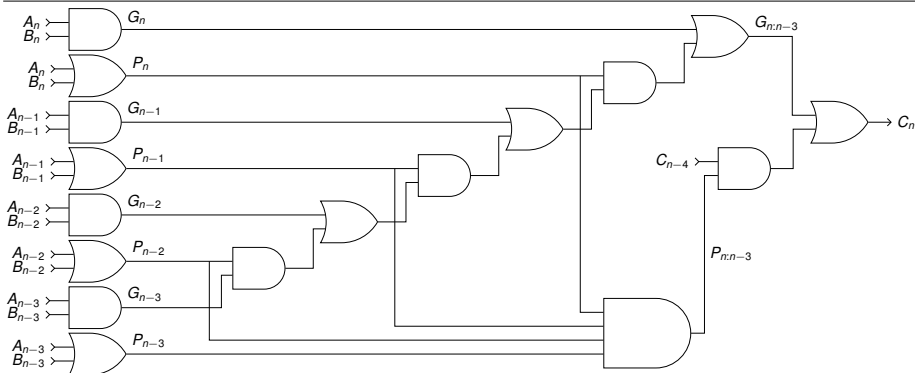
Harris 2013  
Kap. 5.2 + 7.3

# Wiederholung: Schnelle Übertragketten

## Generate und Propagate



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



# Wiederholung: „kleiner als“ Vergleich



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶  $1010_2 < 0011_2 = 0$  für  $u_{2,4}$  bzw. 1 für  $s_2$  Interpretation

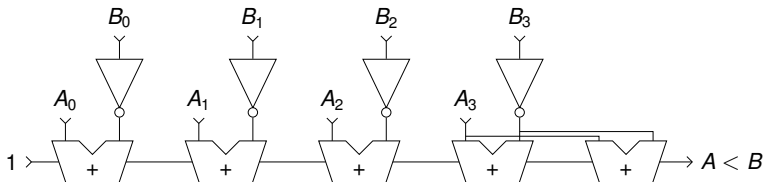
⇒ unterschiedliche Hardware (Vorzeichenexpansion) für Differenz-Bildung bei signed und unsigned nötig

- ▶ für signed:

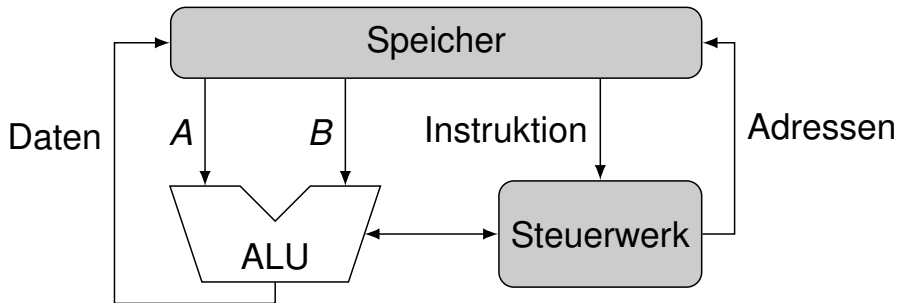
$$\begin{array}{r} 1 \ 0 \ 1 \ 0 \\ - \ 0 \ 0 \ 1 \ 1 \\ \hline \end{array}$$

→

$$\begin{array}{r} 1 \ 0 \ 0 \ 0 \ 1 \\ 1 \ 1 \ 0 \ 1 \ 0 \\ + \ 1 \ 1 \ 1 \ 0 \ 0 \\ \hline = \ 1 \ 0 \ 1 \ 1 \ 1 \end{array}$$

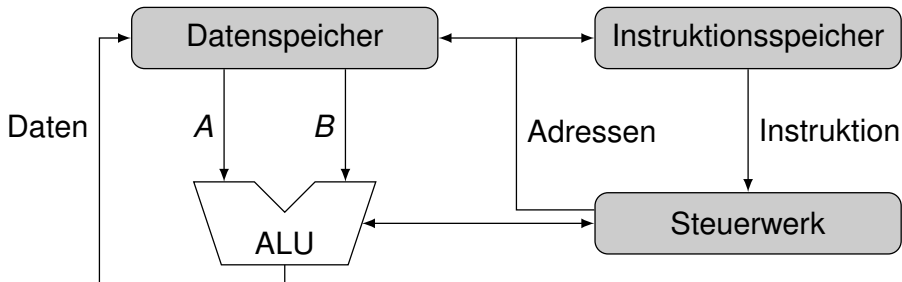


# Wiederholung: Rechnerarchitektur Von-Neumann



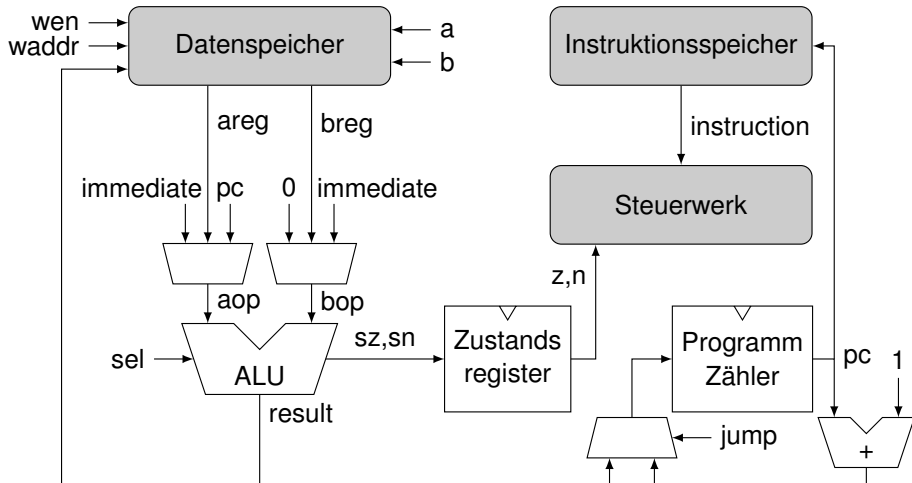


# Wiederholung: Rechnerarchitektur Harvard



# Wiederholung: Rechnerarchitektur

## Harvard Details (Ü13.2)



# Wiederholung: Instruktionssatz für Modellprozessor (Ü13.2)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Befehl	kodierte Instruktion	Registeränderung	nächster PC
ADD(r,a,b)	{4'b0000,7'bx,r,a,b}	$R[r] = R[x] + R[b]$	pc+1
SUB(r,a,b)	{4'b0001,7'bx,r,a,b}	$R[r] = R[a] - R[b]$	pc+1
AND(r,a,b)	{4'b0010,7'bx,r,a,b}	$R[r] = R[a] \& R[b]$	pc+1
OR(r,a,b)	{4'b0011,7'bx,r,a,b}	$R[r] = R[a]   R[b]$	pc+1
XOR(r,a,b)	{4'b0100,7'bx,r,a,b}	$R[r] = R[a] \wedge R[b]$	pc+1
SHL(r,a,b)	{4'b0101,7'bx,r,a,b}	$R[r] = R[a] \ll R[b]$	pc+1
SHR(r,a,b)	{4'b0110,7'bx,r,a,b}	$R[r] = R[a] \gg R[b]$	pc+1
ASHL(r,a,b)	{4'b0111,7'bx,r,a,b}	$R[r] = R[a] \lll R[b]$	pc+1
ASHR(r,a,b)	{4'b1000,7'bx,r,a,b}	$R[r] = R[a] \ggg R[b]$	pc+1
ARED(r,a,b)	{4'b1001,7'bx,r,a,b}	$R[r] = \& R[a]$	pc+1
ORED(r,a,b)	{4'b1010,7'bx,r,a,b}	$R[r] =   R[a]$	pc+1
MOV(r,a)	{4'b1011,7'bx,r,a,0}	$R[r] = R[a]$	pc+1
LDI(immediate)	{4'b1100,immediate}	$R[0] = \text{immediate}$	pc+1
JMP(immediate)	{4'b1101,immediate}		pc+ immediate
JN(immediate)	{4'b1110,immediate}		pc+(n ? immediate : 1)
JZ(immediate)	{4'b1111,immediate}		pc+(z ? immediate : 1)

# Ergänzung: Assembler-Programm

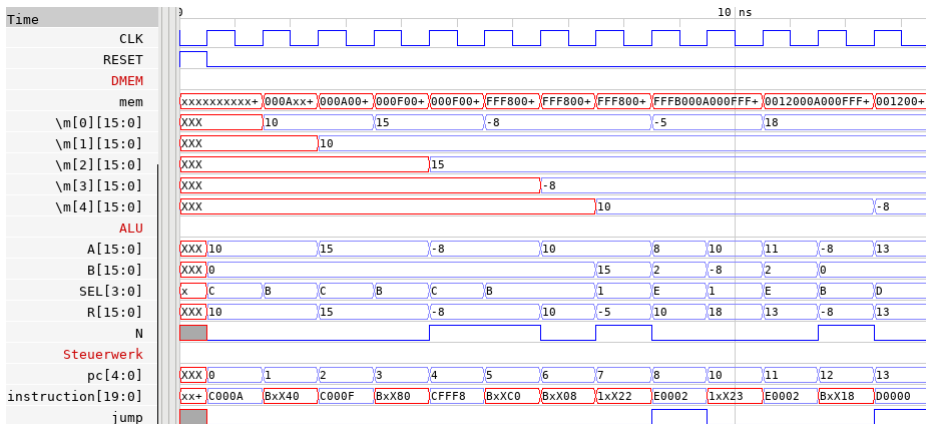
## Beispiel für Minimum von drei Zahlen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

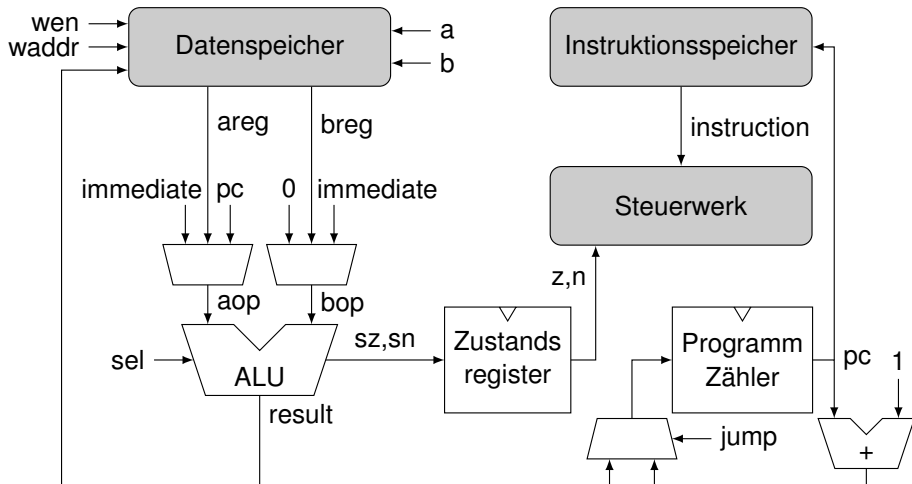
```
1  /*PC*/
2  /* 0*/ LDI( 10);      // R[0] = 10,          N=0, PC=1
3  /* 1*/ MOV(1,0);      // R[1] = R[0] = 10,      N=0, PC=2
4  /* 2*/ LDI( 15);      // R[0] = 15,          N=0, PC=3
5  /* 3*/ MOV(2,0);      // R[2] = R[0] = 15,      N=0, PC=4
6  /* 4*/ LDI( -8);      // R[0] = -8,          N=1, PC=5
7  /* 5*/ MOV(3,0);      // R[3] = R[0] = -8,      N=1, PC=6
8
9  /* 6*/ MOV(4,1);      // R[4] = R[1] = 10,      N=0, PC=7
10
11 /* 7*/ SUB(0,4,2);    // R[0] = R[4] - R[2] = -5, N=1, PC=8
12 /* 8*/ JN(2);         //                      N=0, PC=10
13 /* 9*/ MOV(4,2);      //{R[4] = R[2] = 15      N=0, PC=10}
14
15 /*10*/ SUB(0,4,3);    // R[0] = R[4] - R[3] = 18, N=0, PC=11
16 /*11*/ JN(2);         //                      N=0, PC=13
17 /*12*/ MOV(4,3);      // R[4] = R[3] = -8      N=0, PC=13
18
19 /*13*/ JMP(0);        // Endlosschleife
```

# Ergänzung: Assembler-Programm Simulation der Abarbeitung



# Wiederholung: Rechnerarchitektur

## Harvard Details (Ü13.2)



# Überblick der heutigen Vorlesung



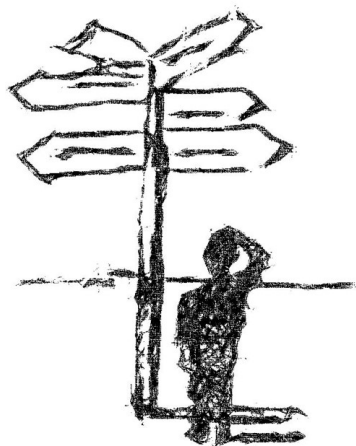
TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Lehrevaluation der FB 20 Fachschaft
- ▶ Field Programmable Gate Arrays
- ▶ Abschluss Digitaltechnik

Anwendungs- software	Programme
Betriebs- systeme	Gerätetreiber
Architektur	Befehle Register
Mikro- architektur	Datenpfade Steuerung
Logik	Addierer Speicher
Digital- schaltungen	UND Gatter Inverter
Analog- schaltungen	Verstärker Filter
Bauteile	Transistoren Dioden
Physik	Elektronen

# Agenda

1. Einleitung
2. Lehrevaluation der FB 20 Fachschaft
3. Field Programmable Gate Arrays (FPGAs)
4. Abschluss Digitaltechnik
5. Zusammenfassung







## ► Ziel / Nutzen

- mittel-/langfristige Verbesserung der Lehre
  - Diskussionsgrundlage für Kontrollgremien des FB 20
  - wird zum Teil der Bewerbungsunterlagen des Dozenten
  - Bewertungsgrundlage für Vergabe von „Preis für gute Lehre“ des FB 20
- ⇒ kommt Studierenden und Lehrenden zugute

## ► Ablauf

- anonymisierte Fragebögen
- ein oder zwei Freiwillige für Einsammeln und Abgabe bei Fachschaft (D120)
- jetzt ausfüllen (später/online nicht möglich)

## Abschnitt 5: Freie Fragen

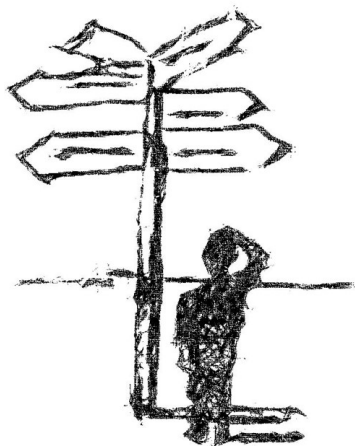


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Haben Sie bereits Erfahrungen mit moderneren didaktischen Konzepten (bspw. Flipped Classroom, Peer Instructions/Feedback, oder E-Teaching) gesammelt?
- ▶ Wären Sie bereit, für den Einsatz modernerer didaktischer Konzepte den Anteil des Selbststudiums zu erhöhen?

# Agenda

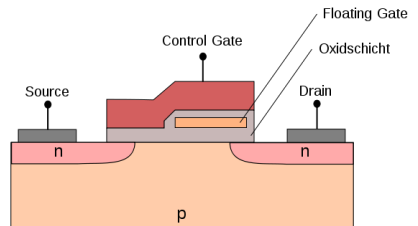
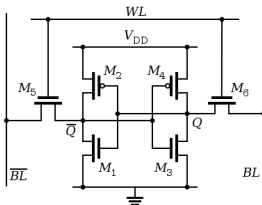
1. Einleitung
2. Lehrevaluation der FB 20 Fachschaft
3. Field Programmable Gate Arrays (FPGAs)
4. Abschluss Digitaltechnik
5. Zusammenfassung





- ▶ Anwendungsspezifische integrierte Schaltungen (ASICs)
  - ▶ führen *für eine Anwendung* optimierte (parallele) Datenpfade aus
  - ▶ Basisgatterschaltungen (bspw. als CMOS) durch optische/chemische Prozesse auf Silikon-Wafer realisiert
  - ⇒ zur Laufzeit nicht an neue Anwendung anpassbar
- ▶ Software-Prozessoren
  - ▶ führen generische Instruktionen sequentiell aus
  - ▶ nur generische (Mikro-)Architektur in Hardware realisiert
  - ⇒ zur Laufzeit durch Austauschen der Instruktionssequenz an neue Anwendung anpassbar
- ⇒ Field Programmable Gate Arrays (FPGAs) vereinen
  - ▶ Flexibilität von Software-Prozessoren („im Feld programmierbar“)
  - ▶ mit Performanz von ASICs (optimierte „Basisgatter-Schaltungen“)

- ▶ FPGAs verwenden feingranulare (bitweise) Konfigurationsspeicher statt wortweise Instruktionspeicher
- ▶ kann mit verschiedenen Speicher-Technologien realisiert werden:
  - ▶ volatil (bspw. SRAM): schnell beschreibbar, benötigt aber permanente Spannungsversorgung (statische Leistungsaufnahme)
  - ▶ nicht-volatil (bspw. Flash): aufwendiger Schreibzugriff, aber Zustand bleibt auch ohne Spannungsversorgung erhalten

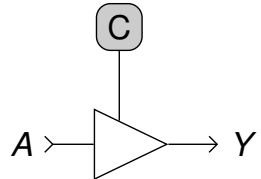
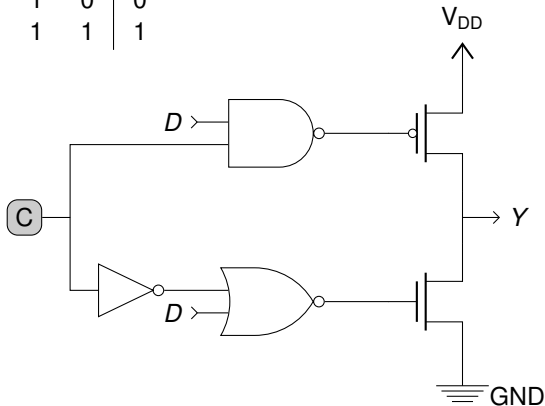


# Programmierbare Schalter



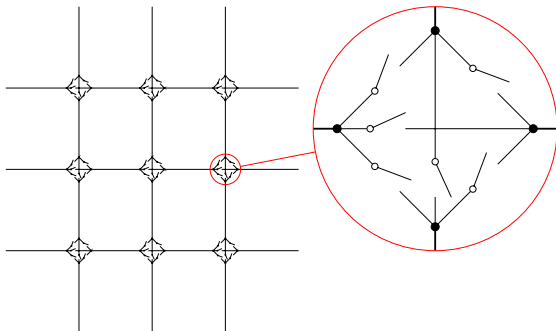
TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$C$	$D$	$Y$
0	*	Z
1	0	0
1	1	1



# Programmierbare Leitungskreuzungen

## Switch Matrix



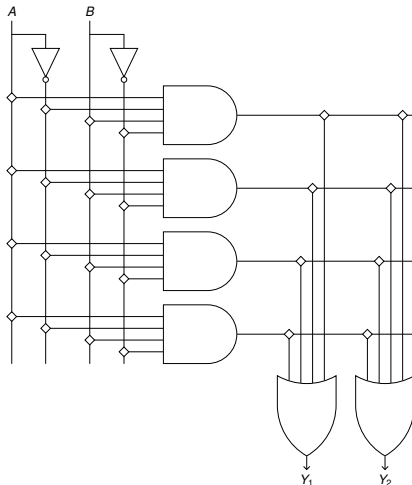
# Programmierbare Logikfelder

## Programmable Logic Array (PLA)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ realisiert kombinatorische Logik
- ▶ zweistufige Logik mit programmierbaren Schaltern in Ein- und Ausgabestufe
- ▶ Varianten:
  - ▶ Programmable ROM: nur Ausgabefeld programmierbar
  - ▶ Programmable Array Logic: nur Eingabefeld programmierbar





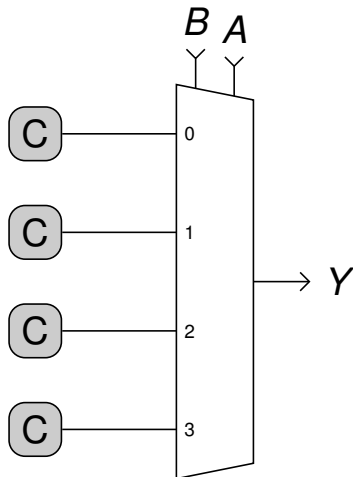
# Programmierbare Tabellen

## Lookup Table (LUT)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ realisiert kombinatorische Logik
- ▶ 2 bis 6 Eingänge
- ▶ häufig auch aufteilbar in kleinere LUTs  
bspw. zwei LUT mit  $n - 1$  gemeinsam  
genutzten Eingängen



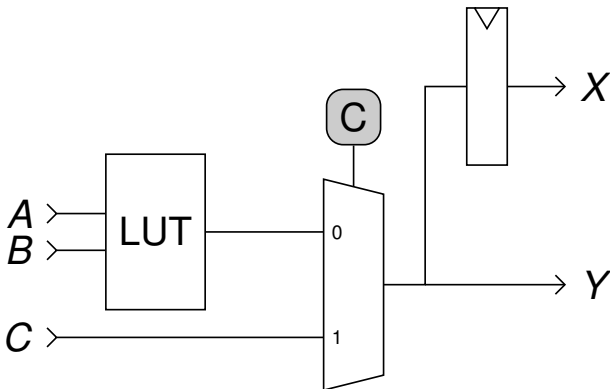
# Programmierbare Logikzelle

## Logic Cell (LC)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

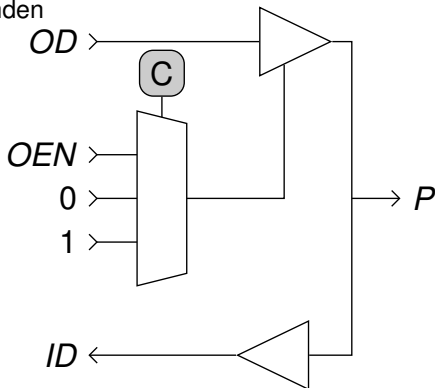
- ▶ kann als kombinatorische Logik und/oder Speicher verwendet werden
- ▶ häufig auch spezielle Carry In/Out für schnelle Arithmetik



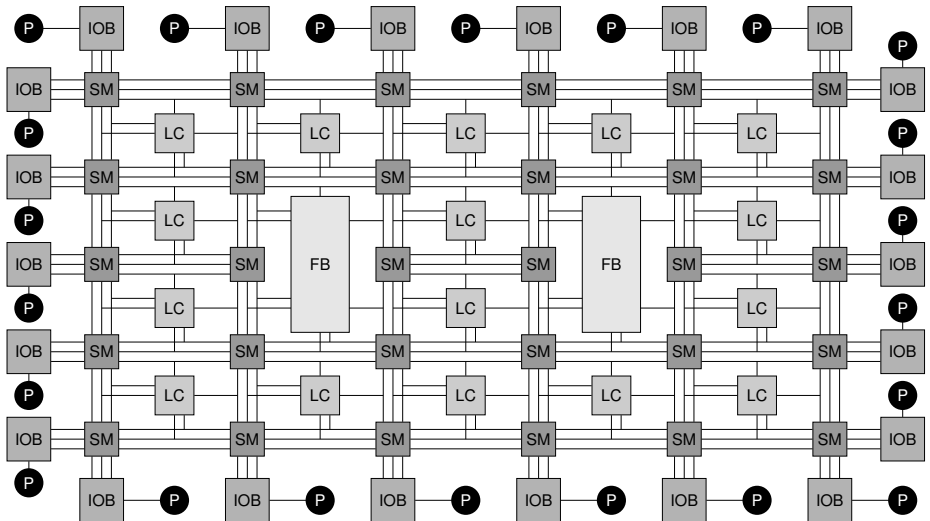
# Programmierbare Ein-/Ausgänge

## Input-/Output Blocks (IOB)

- ▶ Ausgabetreiber kann permanent oder zur Laufzeit steuerbar (*OEN*) deaktiviert werden
- ▶ *P* wird mit physikalischen Pad verbunden
- ▶ häufig auch konfigurierbar:
  - ▶ Spannungs-Level
  - ▶ maximale Stromstärke
  - ▶ Flanken-Steilheit



# Fiel Programmable Gate Array (FPGA)





- ▶ häufig verwendete Logikbausteine als begrenzte Ressourcen verfügbar
  - ▶ Block RAM (BRAM): kleine SRAM Speicher (wenige Kilobit)
  - ▶ Digitale Signalverarbeitung (DSP): Multiplizierer, MAC
  - ▶ Phase-Locked Loop (PLL): Taktmodifikation
  - ▶ Kommunikations-Treiber (USART, USB, Ethernet)
  - ▶ kleine Prozessoren
  - ▶ ...

# Marktrelevante FPGA Hersteller



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Xilinx
  - ▶ Zynq, Virtex, Kintex
  - ▶ 7-series, UltraScale+
- ▶ Intel (hat Altera aufgekauft)
  - ▶ Cyclone, Aria, Stratix
- ▶ Microsemi
  - ▶ IGLOO, SmartFusion, PolarFire, ProASIC
- ▶ Lattice
  - ▶ iCE, Mach

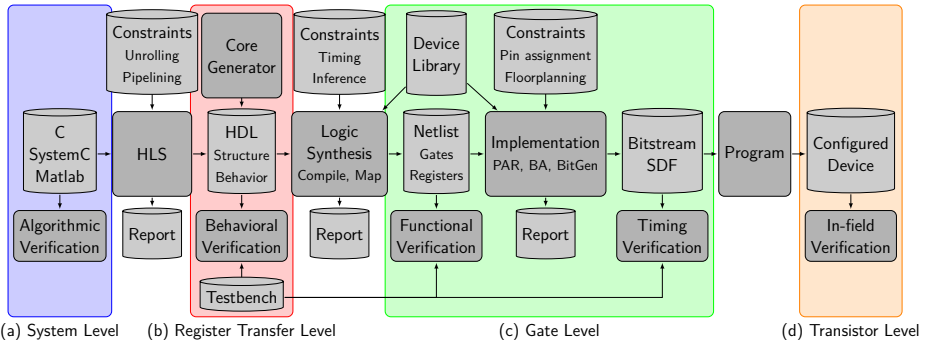


# Xilinx Virtex UltraScale Familie



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

	Device Name	XCVU065	XCVU080	XCVU095	XCVU125	XCVU160	XCVU190	XCVU440
Logic Resources	System Logic Cells (K)	783	975	1,176	1,567	2,027	2,350	5,541
	CLB Flip-Flops	716,160	891,424	1,075,200	1,432,320	1,852,800	2,148,480	5,065,920
	CLB LUTs	358,080	445,712	537,600	716,160	926,400	1,074,240	2,532,960
Memory Resources	Maximum Distributed RAM (Kb)	4,830	3,980	4,800	9,660	12,690	14,490	28,710
	Block RAM/FIFO w/ECC (36Kb each)	1,260	1,421	1,728	2,520	3,276	3,780	2,520
	Block RAM/FIFO (18Kb each)	2,520	2,842	3,456	5,040	6,552	7,560	5,040
	Total Block RAM (Mb)	44.3	50.0	60.8	88.6	115.2	132.9	88.6
Clock Resources	CMT (1 MMCM, 2 PLLs)	10	16	16	20	28	30	30
	I/O DLL	40	64	64	80	120	120	120
	Transceiver Fractional PLL	5	8	8	10	13	15	0
I/O Resources	Maximum Single-Ended HP I/Os	468	780	780	780	650	650	1,404
	Maximum Differential HP I/O Pairs	216	360	360	360	300	300	648
	Maximum Single-Ended HR I/Os	52	52	52	52	52	52	52
	Maximum Differential HR I/O Pairs	24	24	24	24	24	24	24
Integrated IP Resources	DSP Slices	600	672	768	1,200	1,560	1,800	2,880
	System Monitor	1	1	1	2	3	3	3
	PCIe® Gen1/2/3	2	4	4	4	4	6	6
	Interlaken	3	6	6	6	8	9	0
	100G Ethernet	3	4	4	6	9	9	3
	GTH 16.3Gb/s Transceivers	20	32	32	40	52	60	48
Speed Grades	GTY 30.5Gb/s Transceivers	20	32	32	40	52	60	0
	Commercial	—	—	—	—	—	—	-1
	Extended	-1H -2 -3	-1H -2 -3	-1H -2 -3	-1H -2 -3	-1H -2 -3	-1H -2 -3	-2 -3
	Industrial	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2
	Package Footprint <sup>(1, 2, 3)</sup>	Package Dimensions (mm)						
		HR I/O, HP I/O, GTH 16.3Gb/s, GTY 30.5Gb/s						
Footprint Compatible with Kintex® UltraScale Devices	C1517	40x40	52, 468, 20, 20	52, 468, 20, 20	52, 468, 20, 20			
	D1517	40x40		52, 286, 32, 32	52, 286, 32, 32			
	B1760	42.5x42.5		52, 650, 32, 16	52, 650, 32, 16	52, 286, 40, 32		
	A2104	47.5x47.5		52, 780, 28, 24	52, 780, 28, 24	52, 650, 36, 16		
	B2104	47.5x47.5		52, 780, 28, 24	52, 780, 28, 24	52, 780, 28, 24		
	C2104	47.5x47.5		52, 650, 32, 32	52, 650, 32, 32	52, 650, 40, 36	52, 650, 40, 36	
	B2377	50x50		52, 364, 32, 32	52, 364, 40, 40	52, 364, 52, 52	52, 364, 52, 52	
								52, 1248, 36, 0



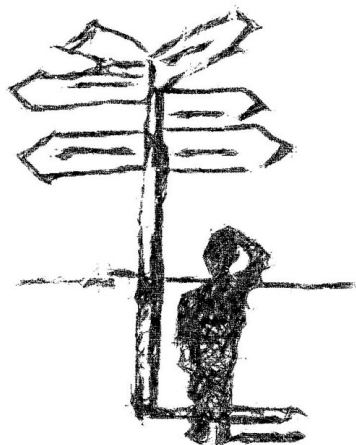


# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

1. Einleitung
2. Lehrevaluation der FB 20 Fachschaft
3. Field Programmable Gate Arrays (FPGAs)
4. Abschluss Digitaltechnik
5. Zusammenfassung



- ▶ *Digitaltechnik*: digitale Abstraktion und ihre technische Umsetzung, Zahlensysteme, Logikgatter, MOSFET Transistoren und CMOS Gatter, Leistungsaufnahme
- ▶ *Kombinatorische Schaltungen*: Boole'sche Gleichungen und Algebra, Abbildung auf Gatter, mehrstufige Schaltungen, vierwertige Logik (0,1,X,Z), Minimierung von Ausdrücken, kombinatorische Grundelemente, Zeitverhalten
- ▶ *Sequentielle Schaltungen*: Latches, Flip-Flops, Entwurf synchroner Schaltungen, endliche Automaten, Zeitverhalten, Parallelität
- ▶ *Hardware-Beschreibungssprachen*: Modellierung kombinatorischer und sequentieller Schaltungen, Strukturbeschreibungen, Modellierung endlicher Automaten, Datentypen, parametrisierte Module, Testrahmen
- ▶ *Grundelemente digitaler Schaltungen*: arithmetische Schaltungen, Fest-/Gleitkommaarstellung, sequentielle Grundelemente, Speicherfelder, Logikfelder

# Aus TUCaN / Modulhandbuch: Qualifikationsziele und Lernergebnisse

- ▶ Studierende **verstehen** nach erfolgreichem Besuch der Veranstaltung die *Konzepte und Grundelemente der digitalen Logik* sowie ihre *technologische Realisierung*.
- ▶ Sie können diese Kenntnisse **selbständig anwenden**, um zielgerichtet *kombinatorische und sequentielle Schaltungen* zu **konstruieren** und in einer *Hardware-Beschreibungssprache* zu **implementieren**.
- ▶ Sie können *digitale Schaltungen* bezüglich *funktionaler und nicht-funktionaler Eigenschaften* **analysieren**.
- ▶ vgl. didaktische Kompetenzhierarchie:  
verstehen → anwenden → analysieren/bewerten → erzeugen

# Wie geht es weiter?

## Vertiefung hardware-naher Themen in



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

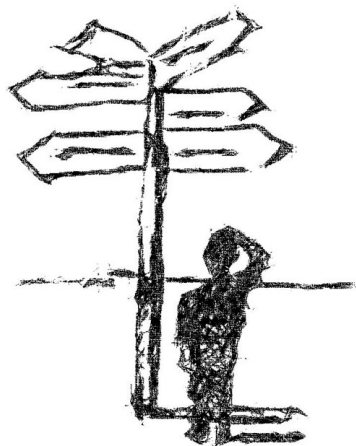
- ▶ Rechnerorganisation
  - ⇒ Prozessorarchitekturen, Befehlssätze, Assemblerprogramme, Mikroarchitekturen, Speicherhierarchie, virtuelle Speicher, Leistungsbewertung
- ▶ Architekturen und Entwurf von Rechnersystemen
  - ⇒ Technologische Trends der Mikroelektronik, Hardware-Entwurfstechniken (mit Bluespec-Verilog), Architekturen für parallele Ausführung, Heterogene Systems-on-Chip, On-Chip und Off-Chip Kommunikationsstrukturen
- ▶ (Fortgeschrittener) Compilerbau
  - ⇒ Hochsprachen-Programme (bspw C, Java) nach Assembler übersetzen, ISA-spezifische Optimierungen (bspw. Registerallokation, Schleifenoptimierung)
- ▶ Embedded-Systems Hands-On
  - ▶ Praxis-naher Einsatz von Mikroprozessoren / FPGAs in kleinen Projekten

# Agenda



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

1. Einleitung
2. Lehrevaluation der FB 20 Fachschaft
3. Field Programmable Gate Arrays (FPGAs)
4. Abschluss Digitaltechnik
5. Zusammenfassung





- ▶ Lehrevaluation der FB 20 Fachschaft
  - ▶ Field Programmable Gate Arrays
  - ▶ Abschluss Digitaltechnik
- 
- ▶ Nächste Vorlesung: Klausurvorbereitung