Vorlesung
Entwurf digitaler Schaltungen

Kapitel 11: Mikroprozessor Grundlagen

Bernhard Rinner

12.09.2024 16:16

UNIVERSITÄT KLAGENFURT

Institut für Vernetzte und Eingebettete Systeme

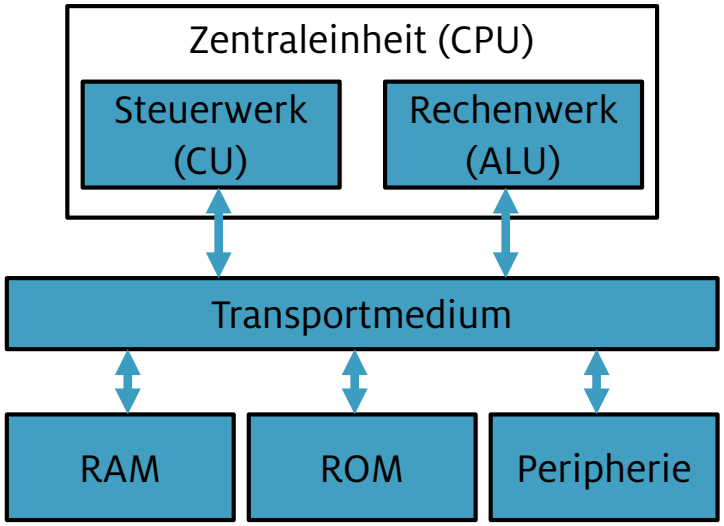
Inhalt

1. Von-Neumann-Architektur [DH: S 352-399]
2. Modellprozessor [MM: S 449-460]
3. Datenpfad [MM: S 461-468]

Von-Neumann-Architektur

Elemente eines Mikrorechners

- Mikroprozessor ist das Herzstück jedes Computersystems
- Von-Neumann-Architektur
  - Zentraleinheit (CPU) aus Rechenwerk und Steuerwerk
  - Rechner ist programmgesteuert. Im Speicher abgelegte Daten werden entsprechend des Befehlssatz interpretiert und ausgeführt
  - Keine Trennung von Programmkode und Daten (im selben Speicher)

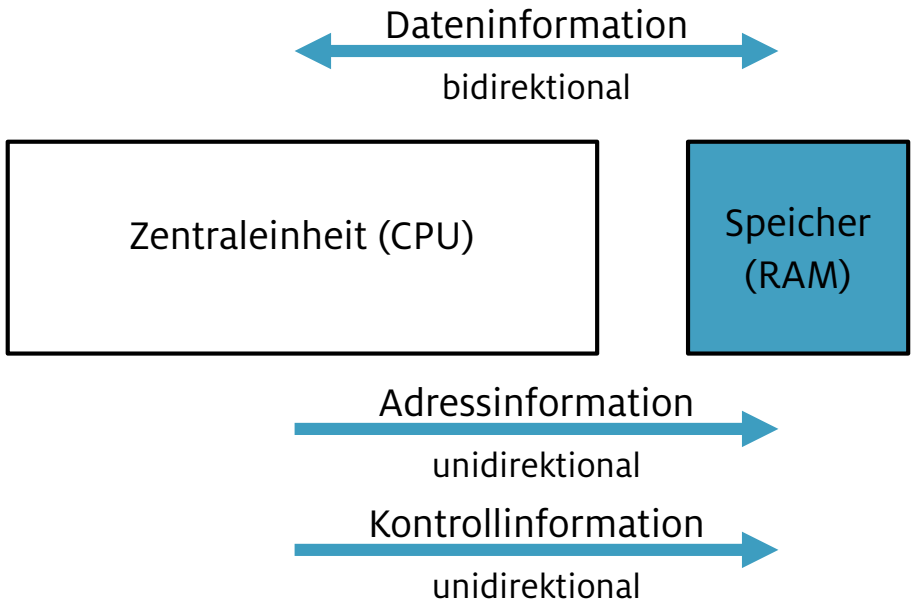


Abarbeitung von Programmen

- Ein Programm besteht aus einer Abfolge von Befehlen, insb.
  - Laden/Speichern von Daten
  - Datenverarbeitung (Inkrement, Dekrement, Addition, Subtraktion etc.)
  - Programmverzweigungen (Sprung, Funktionsaufruf etc.)
- Programme sind im Speicher (häufig ROM) abgelegt und werden nacheinander in Phasen abgearbeitet
  - Befehlsholphase (fetch): Nächster Befehl wird vom Hauptspeicher gelesen
  - Dekodierphase (decode): Im Steuerwerk wird eingelesenes Bitmuster dekodiert und ggf. Operanden nachgeladen
  - Ausführungsphase (execute): Rechenwerk führt Operation aus
  - Speicherphase (write back): Ergebnis wird zurückgeschrieben

Kommunikation CPU und Speicher

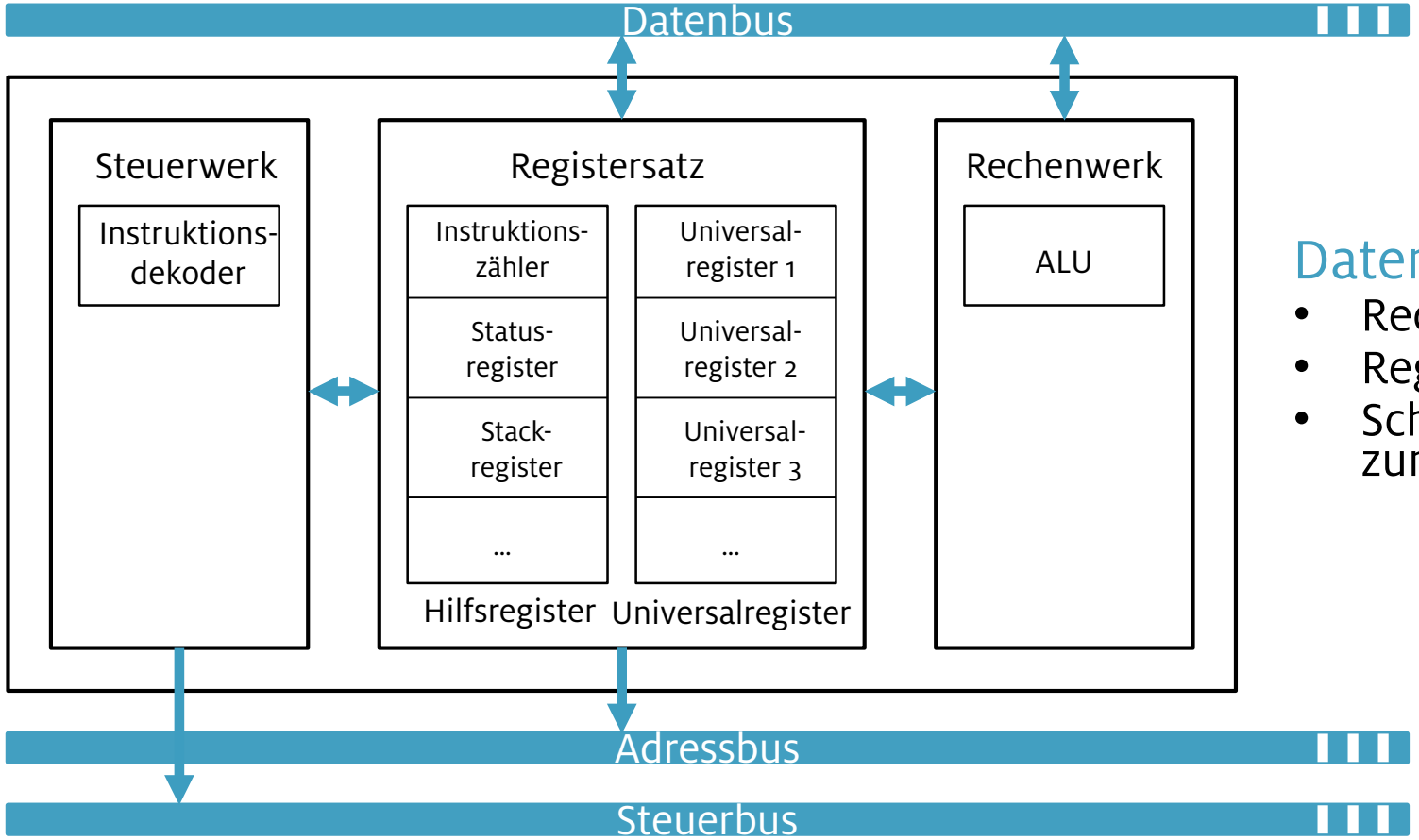
- Zur Kommunikation folgende Informationen erforderlich
  - Adressinformation (zur Identifikation der Speicherzellen)
  - Dateninformation (Inhalt)
  - Kontrollinformation (zur Steuerung des Betriebsmodus)



- Kommunikation mit weiteren Komponenten (ROM, Peripherie)

Aufbau der Zentraleinheit (CPU)

- Aufteilung in Steuerwerk, Registersatz und Rechenwerk



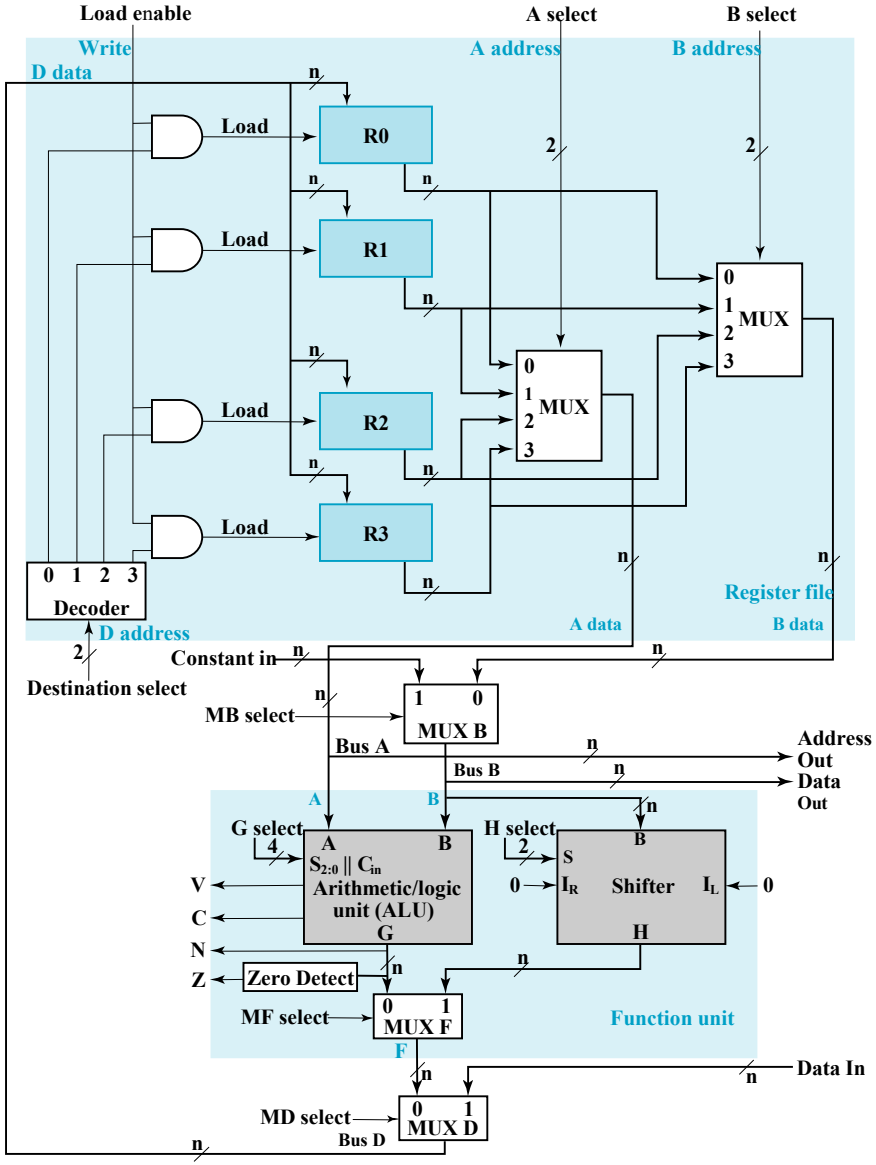
Datenpfad

- Rechenwerk
- Registersatz
- Schnittstelle zum Steuerwerk

Modellprozessor

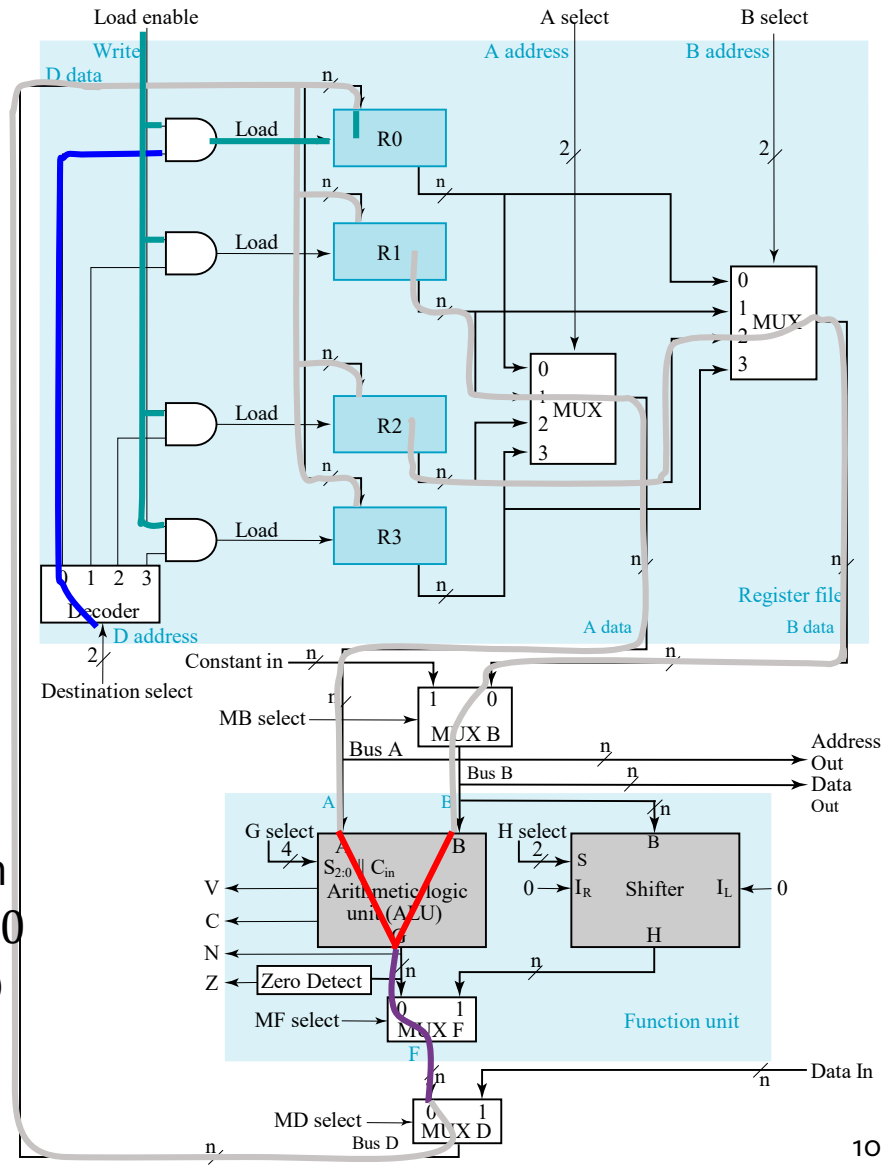
Einfacher Modellprozessor

- Registersatz: Ro, ..., R3
- Auswahl Quellregister über zwei MUX
- MUX B für externe Konstanten
- Daten- und Adressbus mit Bus A und Bus B
- Recheneinheit mit ALU und Schiebeeinheit, MUX F zur Auswahl des Ergebnisses
- MUX D für externen Eingang
- V, C, N, Z Status Bits



Befehlsabarbeitung (Beispiel)

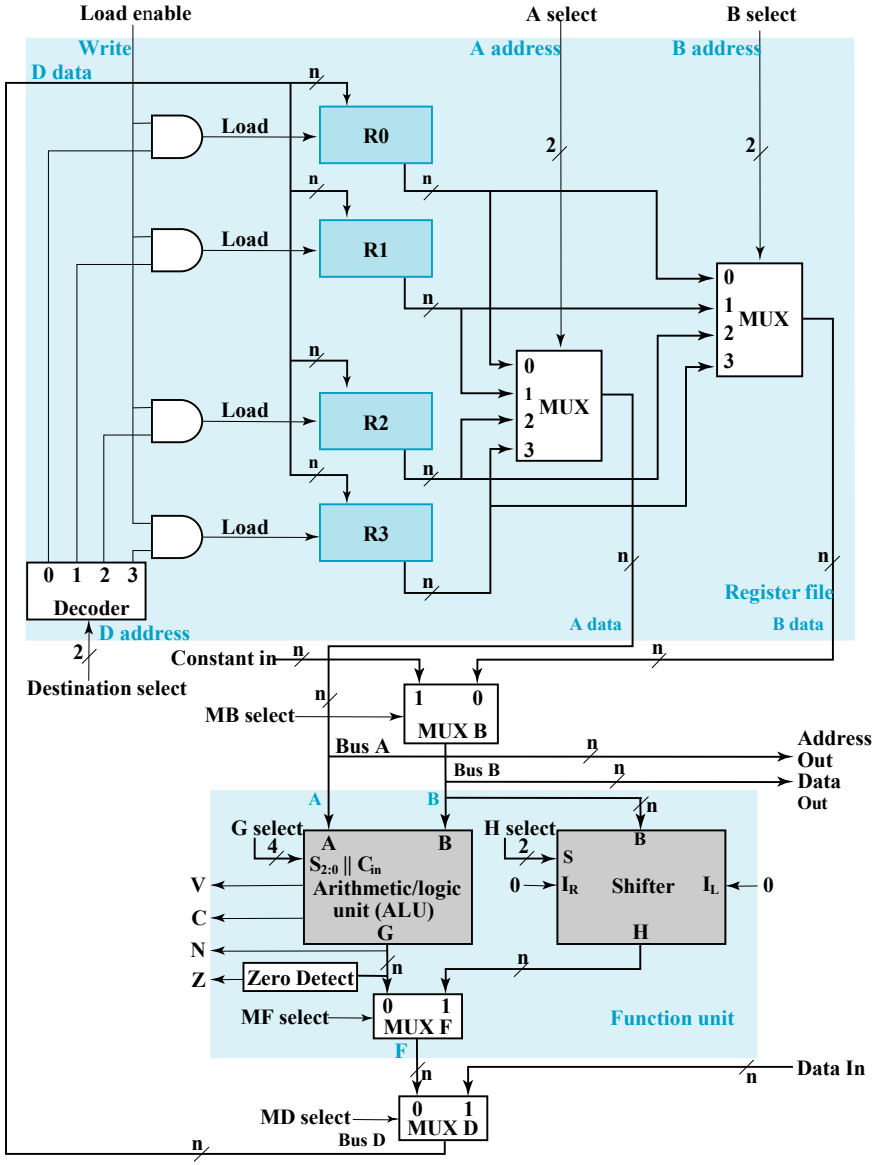
- Mikrooperation: $R0 \leftarrow R1 + R2$
- Auswahl von Inhalt von $R1$ an $A\ data$ durch 01 $A\ address$
- Auswahl von Inhalt von $R2$ an $B\ data$ durch 10 $B\ address$ und Eingabe von 0 an $MB\ select$ um $B\ data$ nach $Bus\ B$ weiterzuleiten
- Eingabe von 0010 an G um Addition $G = Bus\ A + Bus\ B$ auszuführen
- Eingabe von 0 an $MF\ select$ und 0 an $MD\ select$ um den Inhalt von G auf $Bus\ D$ weiterzuleiten
- 00 an Dest$ination\ select$ um Ladeeingang von $R0$ auszuwählen
- Eingabe von 1 an $Load\ enable$ um den Ladeeingang von Ro auf 1 zu setzen ($R0$ wird beim nächsten Takimpuls geladen)
- Gesamte Abarbeitung in 1 Zyklus



Modellprozessor: Steuersignale für alternative Operationen

- Schiebeoperation: 1 an $MF\ select$
- Konstante an Bus B: 1 an $MB\ select$
- Adressen und Daten (Schreiben) für Speicherzugriff: 0 an $Load\ enable$
- Adressen und Daten (Lesen) für Speicherzugriff: 1 an $MD\ select$

- Anmerkung: einige Steuersignale werden nicht berücksichtigt („don't care“)



Realisierung der ALU

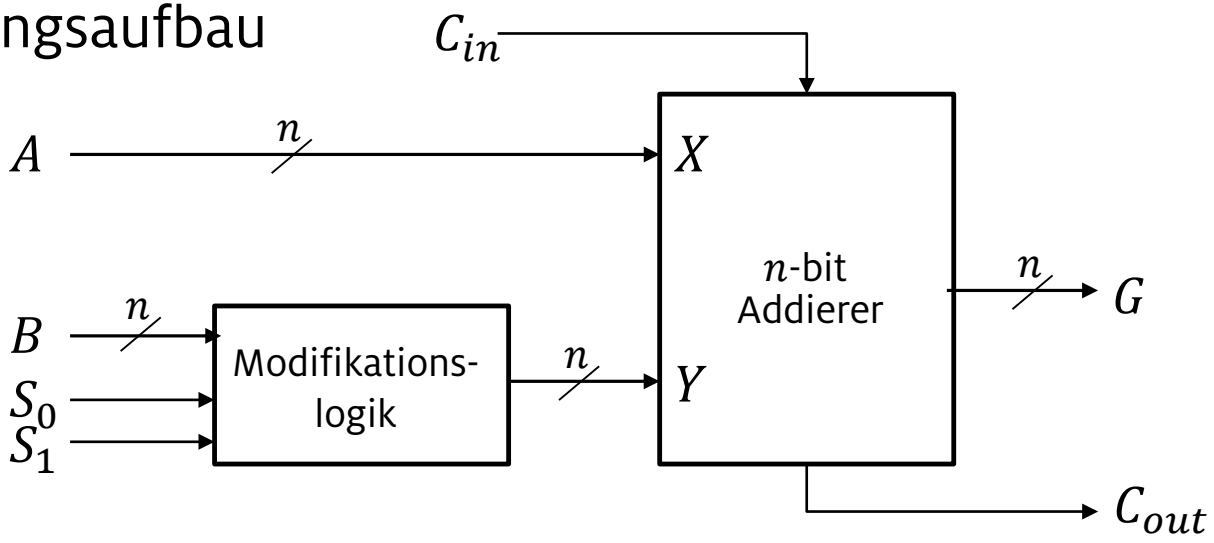
- Unterteilung der ALU in
  - Arithmetische Einheit
  - Logische Einheit
  - Auswahl zwischen den beiden Einheiten
- Entwurf der arithmetischen Einheit durch Unterteilung in
  - n-bit paralleler Addiereinheit
  - Schaltnetz zur geeigneten Modifikation des Eingangs B (n Bits)

Arithmetische Einheit

- Eingang Y aus 4 Modifikationen von B gebildet
  $G = A + Y$

Y	$C_{in} = 0$	$C_{in} = 1$
0	$G = A$	$G = A + 1$
B	$G = A + B$	$G = A + B + 1$
$\bar{B}$	$G = A + \bar{B}$	$G = A + \bar{B} + 1$
1	$G = A - 1$	$G = A$

- Schaltungsaufbau



Arithmetische Einheit

- Funktionstabelle

Auswahl		Eingang		
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	$alle\ 0$	$G = A$ (transfer)	$G = A + 1$ (inkr.)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\bar{B}$	$G = A + \bar{B}$	$G = A + \bar{B} + 1$ (sub)
1	1	$alle\ 1$	$G = A - 1$ (dekr.)	$G = A$ (transfer)

- Anm. Nicht alle Kombinationen erzeugen arithmetisch nützliche Ergebnisse

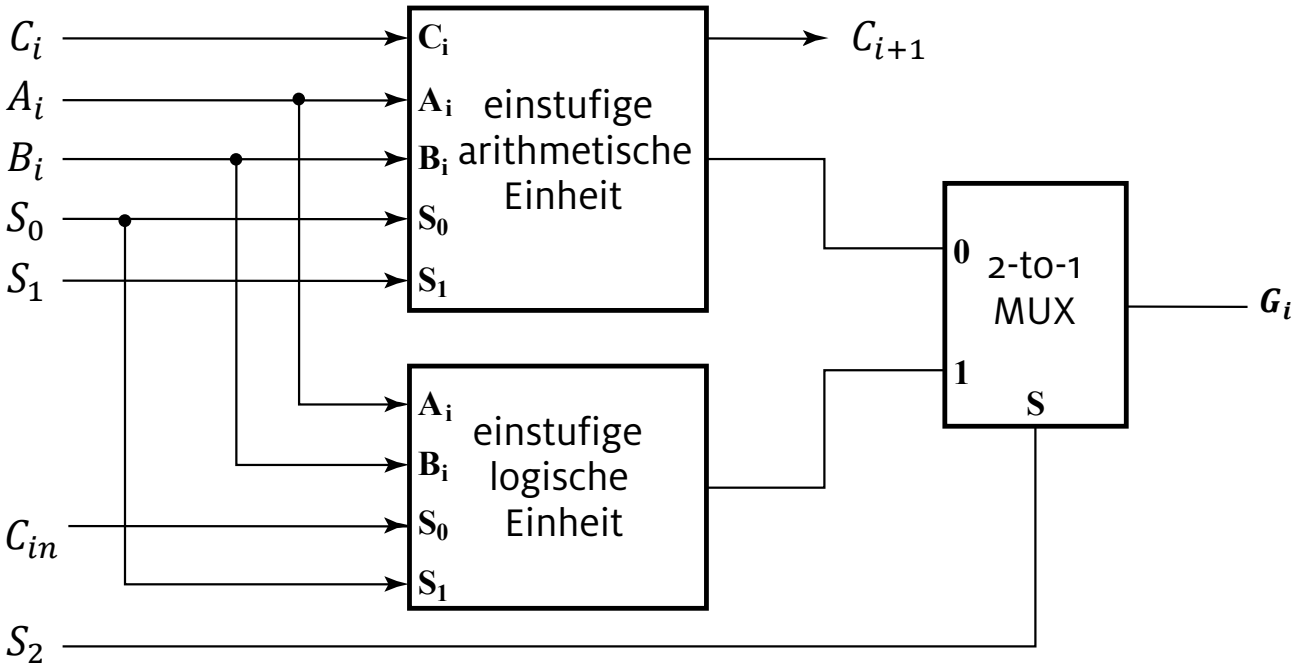
Logische Einheit

- Realisierung der (bitweisen) logischen Verknüpfung UND, ODER, XOR und NICHT
- $(S_1 S_0)$ definiert die Auswahl der logischen Verknüpfung
- Wahrheitstabelle für Bitstelle i (als KV-Diagramm)

- $G_i = S_1 \bar{S}_0 \overline{A_i} \vee S_0 \overline{A_i} B_i \vee \bar{S}_1 A_i B_i \vee S_0 A_i \bar{B}_i$

$S_1 S_0$	UND	$ODER$	XOR	$NICHT$
$A_i B_i$	0 0	0 1	1 1	1 0
0 0	0	0	0	1
0 1	0	1	1	1
1 1	1	1	0	0
1 0	0	1	1	0

Arithmetisch-logische Einheit



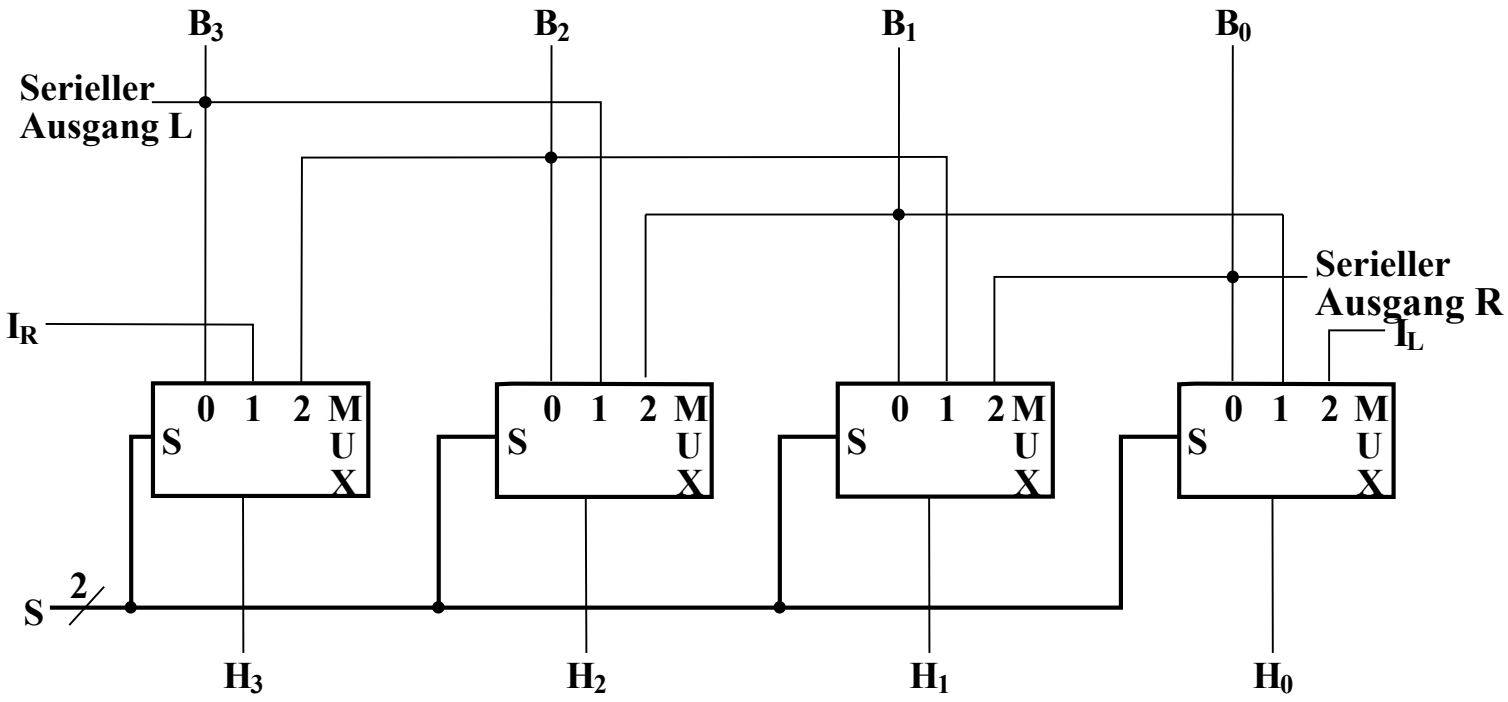
- Aufbau einer einstufigen ALU (für 1 Bit)
  - Carry bei arithmetischer Einheit wird zwischen den Stufen weitergereicht
  - S_2 unterscheidet zwischen logischer/arithmetischer Funktion
  - C_{in} wird mit S_0 der logischen Einheit verbunden

Kombinatorische Schiebeeinheit

UNIVERSITÄT KLAGENFURT

- Verschiebung aller Bits um eine oder mehrere Positionen
  - Richtung: links oder rechts
- Womit sollen die freiwerdenden Stellen aufgefüllt werden?
  - Logisches Schieben: mit 0
  - Arithmetisches Schieben (entspricht Multiplikation mit bzw. Division durch 2):
    beim dem MSB beim Schieben nach links, um das Vorzeichen beizubehalten
  - Rotationsoperation: mit dem „weggeschobenen“ Bit

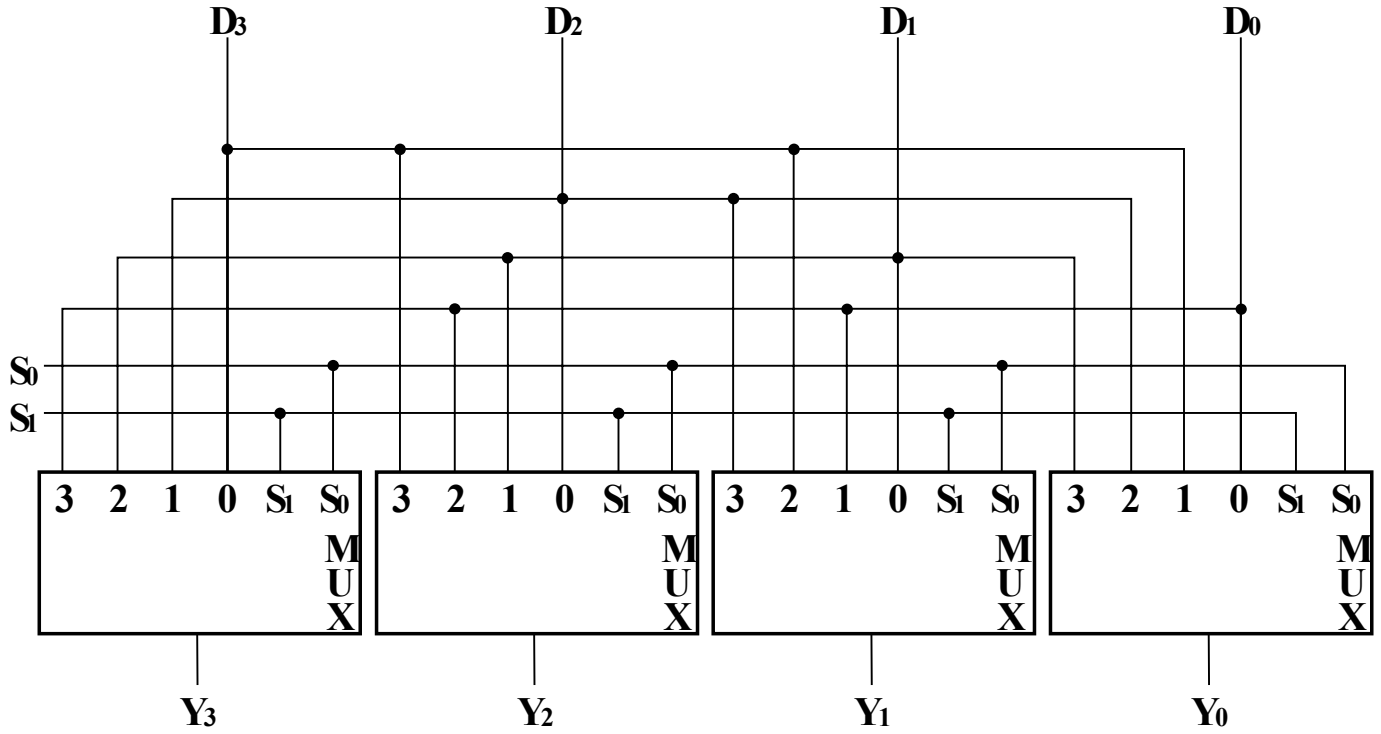
Einfache 4-bit Schiebeeinheit



- Serielle Eingänge
  - I_R bzw. I_L für rechts und links
- Serielle Ausgänge
  - R und L bzw. (MSB- und LSB-Eingang)

Schiebefunktion $(S_1 S_0)$
(00): Transfer
(01): Schieben rechts
(10): Schieben links
(11): nicht verwendet

Barrel Schiebeeinheit



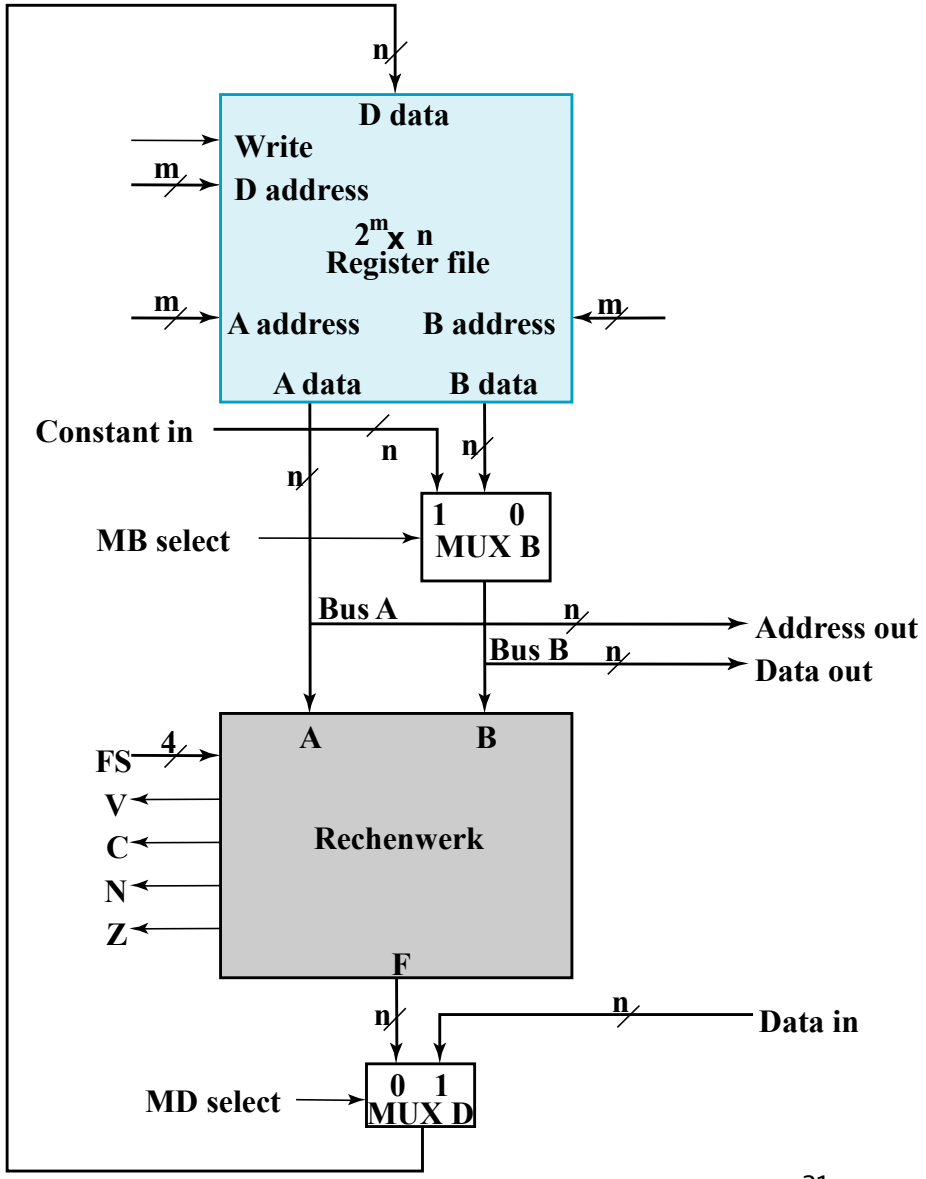
- Kann um mehrere Positionen schieben
  - hier Rotation realisiert

Schiebefunktion $(S_1 S_0)$
(00): Transfer
(01): Schiebe links (1 Pos.)
(10): Schiebe links (2 Pos.)
(11): Schiebe links (3 Pos.)

Datenpfad

Abstraktion des Datenpfad

- Abstraktion von Folie 9
  - Register, Multiplexer, Dekoder und Enable zu Register File
  - ALU, Schiebeeinheit und Multiplexer zu Rechenwerk
  - Verbleibende Busse und Multiplexer werden für Datentransfer benötigt
- Ein/Ausgänge meist selbsterklärend
  - Register file: Load enable zu Write
  - Rechenwerk: FS zur Funktionsauswahl



Funktionsauswahl mit FS

- Definition der Signale $MF\ Select$, $G\ Select$ und $H\ Select$

FS(3:0)	MF Select	G Select(3:0)	H Select(3:0)	Microoperation
0000	0	0000	XX	$F \leftarrow A$
0001	0	0001	XX	$F \leftarrow A + 1$
0010	0	0010	XX	$F \leftarrow A + B$
0011	0	0011	XX	$F \leftarrow A + B + 1$
0100	0	0100	XX	$F \leftarrow A + \bar{B}$
0101	0	0101	XX	$F \leftarrow A + \bar{B} + 1$
0110	0	0110	XX	$F \leftarrow A - 1$
0111	0	0111	XX	$F \leftarrow A$
1000	0	$1X00$	XX	$F \leftarrow A \wedge B$
1001	0	$1X01$	XX	$F \leftarrow A \vee B$
1010	0	$1X10$	XX	$F \leftarrow A \oplus B$
1011	0	$1X11$	XX	$F \leftarrow \bar{A}$
1100	1	$XXXX$	00	$F \leftarrow B$
1101	1	$XXXX$	01	$F \leftarrow sr\ B$
1110	1	$XXXX$	10	$F \leftarrow sl\ B$

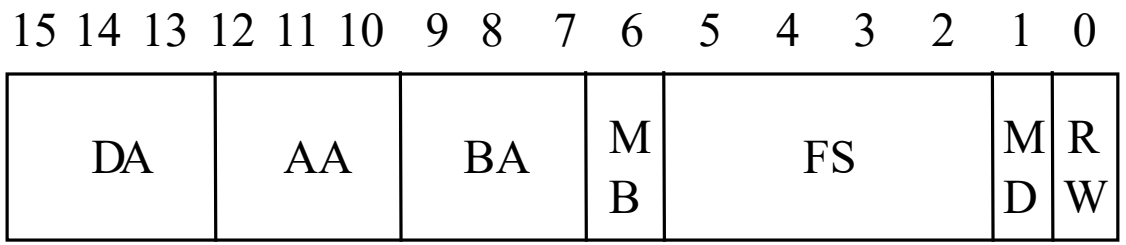
Steuerwort

- Datenpfad hat viele Steuereingänge, die zu einem Steuerwort zusammengefasst werden können

- Die Ausführung der Mikrooperationen kann durch entsprechende Belegung des Steuerwortes vorgegeben werden
- Steuerwort stellt Schnittstelle zu Steuereinheit dar

Steuerwort

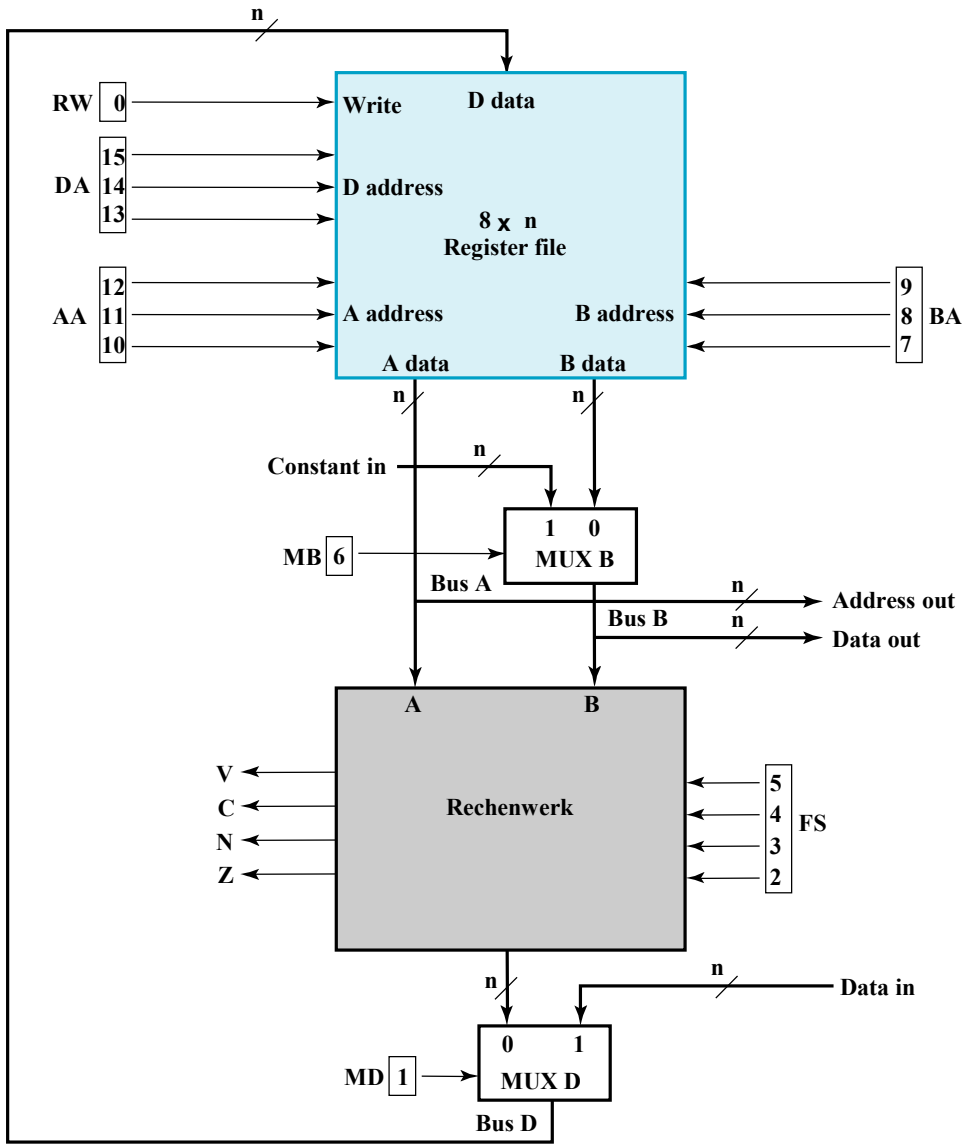
- Format



Steuerwort

- Felder
  - DA – D Address
  - AA – A Address
  - BA – B Address
  - MB – Mux B
  - FS – Function Select
  - MD – Mux D
  - RW – Register Write

Steuerwort und Datenpfad



Steuerwort: Kodierung

Encoding of Control W

DA, AA, BA		MB		FS		MD		RW	
Function	Code	Function	Code	Function	Code	Function	Code	Function	Code
$R0$	000	Register	0	$F \leftarrow A$	0000	Function	0	No write	0
$R1$	001	Constant	1	$F \leftarrow A + 1$	0001	Data In	1	Write	1
$R2$	010			$F \leftarrow A + B$	0010				
$R3$	011			$F \leftarrow A + B + 1$	0011				
$R4$	100			$F \leftarrow A + \bar{B}$	0100				
$R5$	101			$F \leftarrow A + \bar{B} + 1$	0101				
$R6$	110			$F \leftarrow A - 1$	0110				
$R7$	111			$F \leftarrow A$	0111				
				$F \leftarrow A \wedge B$	1000				
				$F \leftarrow A \vee B$	1001				
				$F \leftarrow A \oplus B$	1010				
				$F \leftarrow \bar{A}$	1011				
				$F \leftarrow B$	1100				
				$F \leftarrow sr\ B$	1101				
				$F \leftarrow sl\ B$	1110				

Mikrooperationen: symbolische Darstellung

Mikro-operation	DA	AA	BA	MB	FS	MD	RW
$R1 \leftarrow R2 - R3$	$R1$	$R2$	$R3$	**Register**	$\mathrm{F} \leftarrow A + \bar{B} + 1$	**Function**	**Write**
$R4 \leftarrow sl\ R6$	$R4$	—	$R6$	**Register**	$\mathrm{F} \leftarrow sl\ B$	**Function**	**Write**
$R7 \leftarrow R7 + 1$	$R7$	$R7$	—	**Register**	$\mathrm{F} \leftarrow A + 1$	**Function**	**Write**
$R1 \leftarrow R0 + 2$	$R1$	$R0$	—	**Constant**	$\mathrm{F} \leftarrow A + B$	**Function**	**Write**
$Data\ out \leftarrow R3$	—	—	$R3$	**Register**	—	—	**No Write**
$R4 \leftarrow Data\ in$	$R4$	—	—	—	—	**Data in**	**Write**
$R5 \leftarrow 0$	$R5$	$R0$	$R0$	**Register**	$\mathrm{F} \leftarrow A \oplus B$	**Function**	**Write**

Mikrooperationen: binäre Darstellung

Mikro-operation	DA	AA	BA	MB	FS	MD	RW
$R1 \leftarrow R2 - R3$	001	010	011	0	0101	0	1
$R4 \leftarrow sl\ R6$	100	XXX	110	0	1110	0	1
$R7 \leftarrow R7 + 1$	111	111	XXX	X	0001	0	1
$R1 \leftarrow R0 + 2$	001	000	XXX	1	0010	0	1
$Data\ out \leftarrow R3$	XXX	XXX	011	0	$XXXX$	X	0
$R4 \leftarrow Data\ in$	100	XXX	XXX	X	$XXXX$	1	1
$R5 \leftarrow 0$	101	000	000	0	1010	0	1

Bernhard Rinner
E: bernhard.rinner@aau.at
W: http://nes.aau.at

UNIVERSITÄT KLAGENFURT
Institute für Vernetzte und Eingebettete Systeme