



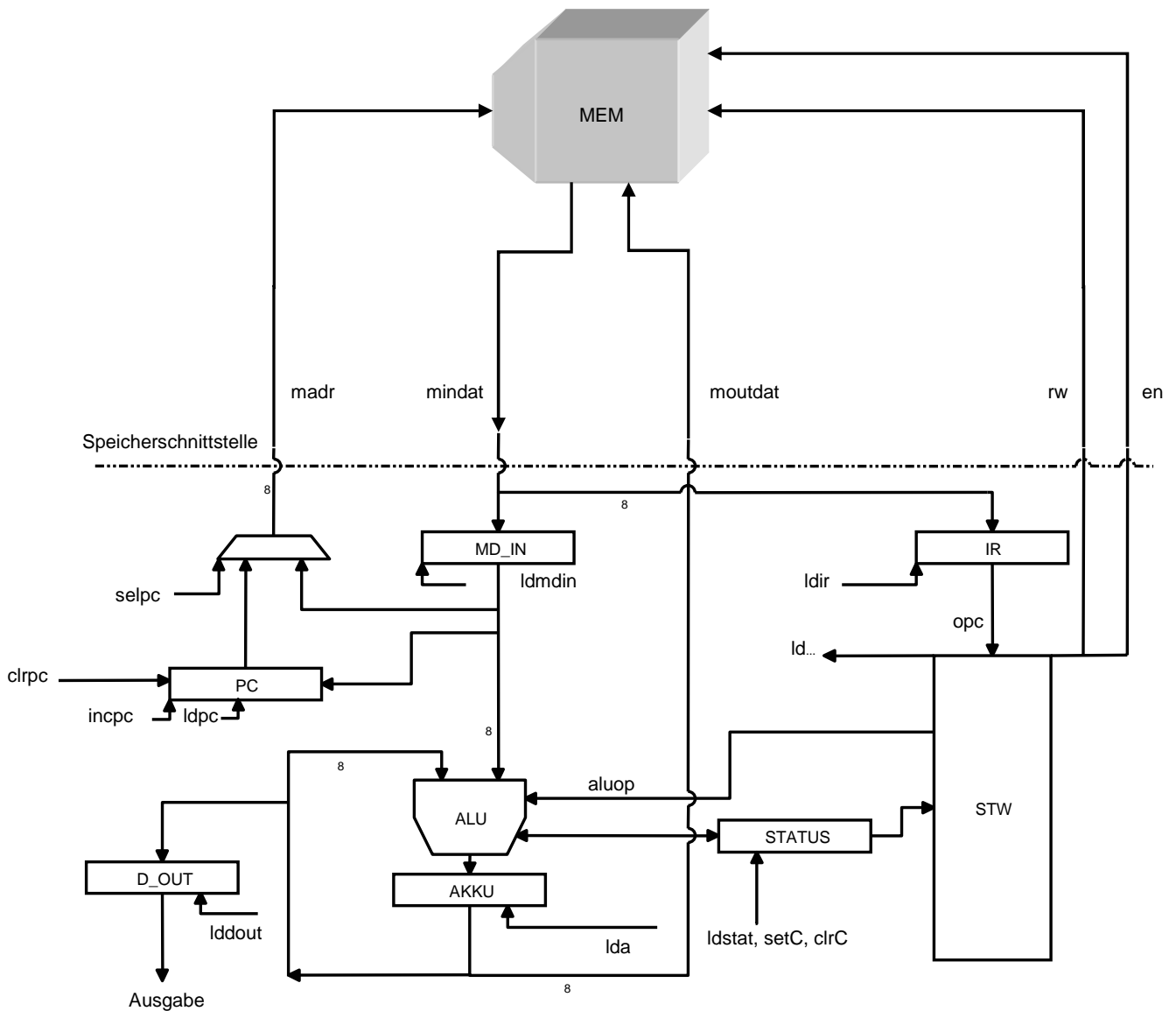
## 7. Aufgabenblatt

02.06.2010

### Aufgabe 1: Realisierung der ALU für den Modellrechner

Die in den Folien der 7. Vorlesung spezifizierte ALU soll nun in Verilog HDL umgesetzt werden.

Die folgende Abbildung zeigt nochmals die Architektur/Struktur des Modellrechners.



Die Liste der Maschinenbefehle des Modellrechners sind in der folgenden Tabelle angegeben. Alle Befehle der ALU und die dadurch veränderten Statusbits sind der Liste zu entnehmen. Statusbits sind: O-Overflow, C-Carry, N-Negativ und Z-Zero.

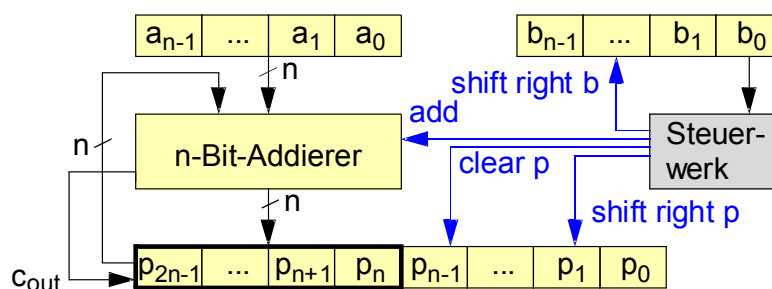
**Status:** Beim Statusbits bedeutet ! = *wird verändert* und + = *wird nicht verändert*.  
Der Eintrag – bedeutet, dass die entsprechenden Befehle keine Statusbits beeinflussen.

OPC	I	Befehl	MNEMONIC	Byte1	Byte2	Status OCNZ
0	0	lade AKKU mit Wert	LDA #C	0000 0000	WERT	00!!
0	1	lade AKKU mit dem Speicherinhalt von Adresse	LDA N	0000 0001	ADRESSE	00!!
1	X	AKKU bitweise negieren	NOT	0001 000x		00!!
2	0	AKKU mit Wert verunden	AND #C	0010 0000	WERT	00!!
2	1	AKKU mit dem Speicherinhalt von Adresse verunden	AND N	0010 0001	ADRESSE	00!!
3	0	AKKU mit Wert verodern	OR #C	0011 0000	WERT	00!!
3	1	AKKU mit dem Speicherinhalt von Adresse verodern	OR N	0011 0001	ADRESSE	00!!
4	0	Wert zum AKKU addieren	ADD #C	0100 0000	WERT	!!!!
4	1	Speicherinhalt von Adresse zum AKKU addieren	ADD N	0100 0001	ADRESSE	!!!!
5	X	AKKU nach rechts schieben	SHR	0101 000x		0!!!
6	X	AKKU nach rechts schieben Vorzeichen nachziehen	ASR	0110 000x		0!!!
7	X	AKKU nach links schieben	SHL	0111 000x		0!!!
8	X	Carrybit im Statusregister löschen	CLRC	1000 000x		+0++
9	X	Carrybit im Statusregister setzen	SETC	1001 000x		+1++
10	X	NOF, keine Operation	NOF	1010 000x		–
11	X	Sprung zur Adresse Label	BRA L	1011 0000	LABEL	–
12	X	Sprung zur Adresse Label falls Z-Statusbit gesetzt ist	BRZ L	1100 0000	LABEL	–
13	X	Schreibe Inhalt des AKKUs an Adresse in den Speicher	STA N	1101 0000	ADRESSE	–
14	X	Schreibe Inhalt des AKKUs in das Ausgaberegister	OUT	1110 000x		–

- a) Implementieren Sie die ALU als Schaltnetz in Verilog HDL. Die vorzeichenbehafteten Zahlen werden als 2K-Zahlen dargestellt. Testen Sie die korrekte Funktionsweise der Befehle.

## Aufgabe 2: Implementierung einer seriellen Multiplikation

Folgendes Schaltwerk realisiert eine serielle Multiplikation.



---

Der Algorithmus zur seriellen Multiplikation funktioniert wie folgt.  
Die Befehle in Klammern geben die Mikrooperationen an (vgl. Implementierung in Verilog HDL).

- Lösche p (clear p)
- Ermittle  $b_0$
- Addiere a auf  $(p_{2n-1}, \dots, p_{n+1}, p_n)$  oder nicht, je nach  $b_0$  (add)
- Verschiebe p einschließlich  $c_{out}$  der vorherigen Addition (shift right p)
- Verschiebe b (shift right b)

- a) Berechnen Sie nach dem vorgestellten Algorithmus das Produkt  $p = a \cdot b$  für  $a = 0010$  und  $b = 0011$ .
- b) Implementieren Sie die vorgestellte serielle Multiplikation in Verilog HDL. Die Register a und b sind jeweils 4 Bit breit. Das Ergebnisregister p ist 8 Bit breit. Verwenden Sie die vorgegebenen Mikrooperationen. Das Steuerwerk kann als Zustandsgraph beschrieben werden. Die Kodierung der Zustände kann frei gewählt werden.
- c) Zeigen Sie die korrekte Funktionsweise durch eine Simulation.