

TI2 13.6.2012

noch zu „Erweiterungen der Prozessorgrundstruktur“

→ durch strukturelle Erweiterungen ist ein verallgemeinerter Befehlsablauf mgl.

→ F 3_130 als PN

enthält alle möglichen Befehlsvarianten auf abstrakten Niveau

1. Minimale Variante:

→ kein Operanden lesen und schreiben aus dem Speicher, Operanden nur in internen Registern bzw. es werden keine benötigt

- (1) Befehl lesen und dekodieren (BL&BD)
- (2) Befehl ausführen (BA)
- (3) Erzeugen der nächsten Befehlsadr. (ENBA)

2. Mittlere Variante (entspricht F 3_60, Bsp., andere mgl.)

- (1) BL&DK
- (2) 1. Operand (1. OL) lesen
- (3) BA
- (4) ENBA

3. Maximale Variante

→ alle Operanden einschließlic Resultat von und zum Speicher

- (1) BL&DK
- (2) 1. OL
- (3) 2. OL
- (4) BA
- (5) RESS (OS) (Resultat schreiben)
- (6) ENBA

3.3. Befehlsübersicht

(hier allgemeine Übersicht, nicht auf konkreten Prozessor bezogen)

→ in der Übung i86 Architektur, flaches Programmiermodell

3.3.1. Datentransferbefehle

- Veränderung des Ortes von Operanden
- i.a. im Kontext vorheriger bzw. späterer Befehle (z.B. Vorbereitung einer Ausführung)

Ziel, Quelle:

- ORi, PSR (**Transport**)
- Speicherplätze (evtl. struktuiert) (**Transport**)
- Ein-/Ausgabeschnittstellen (**Ein-/Ausgabe**)
- OA (**Transport**)
- Stack (**Stackbefehle**)

evtl. nicht alle Kombinationen implementiert

Bsp. (fiktiver Assembler):

MOVE OR1, (123456789ABCDEFh)

Transport Operand von der Speicherzelle mit der Adresse 1...Fh zum OR1

Transportprinzip ist zumeist Kopieren:

Ziel wird mit Inhalt der Quelle überschrieben, Inhalt der Quelle bleibt erhalten

3.3.2. Arithmetik- und Logikbefehle

➔ Operandenbehandlung in der ALE

- Operationen (das ist im Weiteren die Ausprägung von schwarz)

a) Grundrechenarten (Addition, Subtraktion, Multiplikation, Division) (in den im Prozessor verfügbaren Datenformaten) (z.B. Byte, 2 ..8 Byte in Zweierkomplement, unvorzeichenbehaftetes Festkomma, Gleitkomma)

-> i.a. nur max. 2 E- und ein A-Operand, mehr müssen in Befehlsfolgen behandelt werden

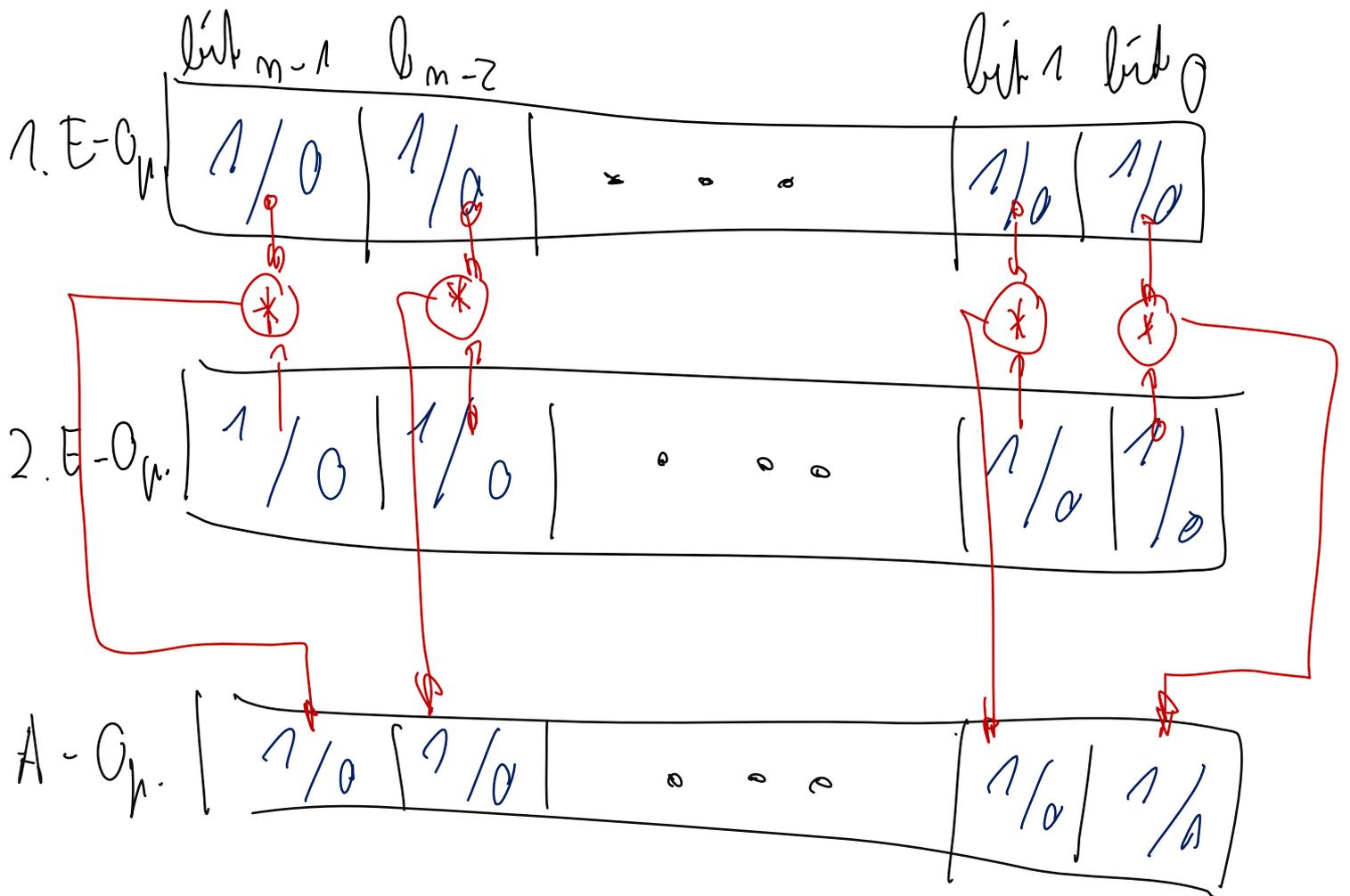
- höhere Operationen werden i.a. durch Befehlsfolgen realisiert

- Befehlsfolgen zumeist als vorgefertigte (Arithmetik-) Bibliotheken

b) Logische Grundoperationen

binäre Logik, max. 2 E-Op. 1 A-Op.

Datendarstellung: Zusammenfassung von n Bitvariablen (n=8, 16 ...)



(*) : UND, ODER, Exklusiv ODER
 Negation (nur 1 E-Op.!))

Gesümdertes Ergebnis

$$y_3 = x_{27} \wedge x_3$$

y_3 - bit₃ (A-Op.)

$$X_{27} = \text{bit}_{27} (E - O_{n. 1})$$

$$X_3 = \text{bit}_3 (E - O_{n. 2})$$

Vor- und Nachbeweiser notwendig
(Befehle im Weiteren)

$$\text{bit}_{27} (E - O_{n. 1}) \rightarrow \text{bit}_3 \rightarrow E - \text{Reg. 1}$$

$$\text{bit}_3 (E - O_{n. 2}) \rightarrow \text{bit}_3 (E - O_{n. 2} = E - \text{Reg. 2})$$

UND

$$\text{bit}_3 (A - \text{Reg}) = \text{bit}_3 (A - O_n)$$

Bsp. ADD OR18, (Index-Reg.+Indexoffset)

Addition des Operanden in OR18, mit dem Inhalt der Speicherzelle, deren Adresse aus Index-Reg-Inhalt+Indexoffset-Inhalt gebildet wird, nach Operation wird Indexoffset Inhalt verändert (siehe Kap. 3.2.), Ergebnis in OR18

→ Kombination Ausführung und Datentransport

Bsp. AND OR3; OR2,OR5

OR3 enthält nach der Operation die bit-weise UND-Verknüpfung aller bits von OR2 und OR5

nächstes mal: weiter mit Vergleich und Test