

Weiter zu 3.2.2 Ablauf von Befehlen in der Grundstruktur

Ablauf in Schritten:

1. BA-Inhalt (BAD: 100h) auf externen Adressbus, parallel dazu erzeugt die AST das Signal SPL (Speicher lesen), dies wird auf dem externen Steuerbus ausgegeben
2. Mit Adressbusinhalt und SPL bildet der Speicher den aktuellen Befehlscode (SUB, 03h), der wird über den Datenbus zum BR transportiert.
3. BR-Inhalt kommt zum Befehlsdeko­der. Der dekodiert ihn und erzeugt die für den aktuellen Befehl gültige BV (Befehlsvariante: die Variante, die aktuell ausgeführt werden und deren logisch zeitliches Verhalten.
4. OA-Inhalt (200h) auf externen AB, AST verhält sich wie beim Beheht lesen.
5. Speicher gibt Operanden (2h) zurück, der kommt ins OR2
6. BD liefert zur ALE die Steuersignale zur aktuellen Operation (SUB, OPN), Eingangsoperanden kommen von OR1 und OR2 (2h, 2h)
7. ALE führt OPN aus, Ergebnis (ERG, 0) kommt zum OR1, PSR erhält den Prozessorstatus, der dem Befehl erzeugt wurde (z-bit ist gesetzt)
8. Erzeugen der Befehlsadresse für den nächsten Befehl durch $BA := BA + 1$ (101h): Vorbereitung des Startens des Folgebefehls.

3.3. Erweiterungen der Prozessorgrundstruktur

- Erweiterung auf mehr als zwei OR_i, die je nach Befehlscode OP1, den OP2 oder das ERG enthalten. Vorteil: Es viel seltener notwendig, Operanden im Speicher zu schreiben, d.h. der Prozessor arbeitet schneller (F3_70)
- Erweiterte Strukturen der Operanden- und Befehlsadresse

Grundidee: zweiteilige Adresse

- Erster Teil adressiert das Teilprogramm (1. Befehl des TP)
- Zweiter Teil adressiert den Abstand eines aktuellen Befehls zur Adresse des ersten Befehls des Teilprogramms

Physische Realisierung:

Zwei Adressregister, das erste (Basis-OA) für die Adresse des ersten Befehls der TP, das zweite (Offset-OA) für den Abstand für den ersten Befehl des TP, $\text{Gesamt-OA} = \text{Basis-OA} + \text{Offset-OA}$

(es sind auch andere Möglichkeiten der Bereitstellung der beiden OA-Adressteile vorhanden)

Basisadressierung für Operanden (F3_80 links) und für Befehle in F3_100.

Index-Basisadressierung Grundidee:

- Zweiteilige Adresse
- Erster Teil adressiert den Anfang (oder das Ende einer regelmäßig aufgebauten Datenstruktur (z.B. Vektor, regelmäßig, weil auf das aktuelle Element folgt immer das im Speicher anschließende Element)

Zweiter Teil adressiert den Abstand eines aktuellen Elements gegenüber dem ersten (bzw. letztem Element bei 3.2.2 Ablauf von Befehlen in der Grundstruktur

- „Rückwärtsarbeitsweise“, das heißt, auf das aktuelle Element folgt immer das vorherige)
- Abstand zwischen einem aktuellen Element zu seinem Nachfolger (bzw. Vorgänger) ist konstant
- Ablauf ähnlich Basisadressierung, zusätzlich erfolgt nach einem Speicherzugriff auf ein Element die Addition (bzw. Subtraktion) von dem Abstand (n) zum (bzw. vom) des Offsets der aktuellen Adresse eines Elements (F3_80 rechts)

Basisadressierung und Indexadressierung können gekoppelt vorkommen.

Bsp. zur Basisadressierung für zwei TP in F3_90.