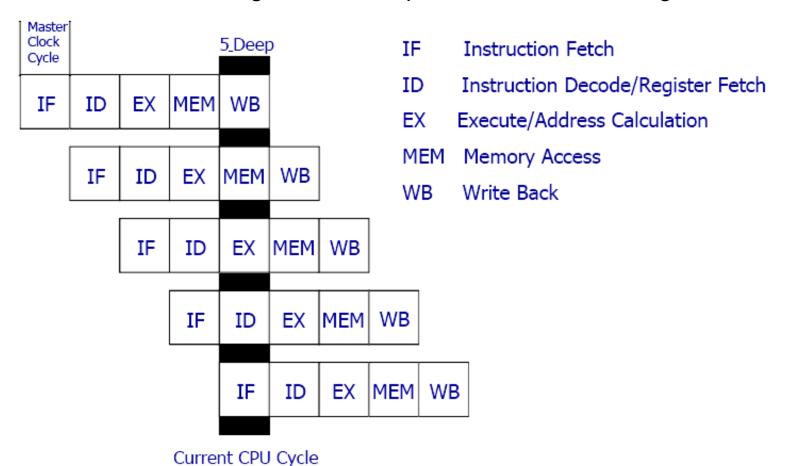
Übung 6

Pipeline - Verarbeitung

Pipeline - Konflikte

Übung 6

Betrachtet werde die folgende DLX- Pipeline aus der Vorlesung:



Übung 6 – Aufgabe 1

Gegeben sei das folgende sequenzielle Programmstück, wobei a und b Speicheradressen darstellen:

```
S1: LOAD R1, a ; R1:= [a]

S2: LOAD R2, b ; R2:= [b]

S3: ADD R2,R1,R2 ; R2:= R1 + R2

S4: MUL R1,R1,R2 ; R1:= R1 * R2
```

- 1. Welche Datenabhängigkeiten bestehen im Programm?
- 2. Wie viele Pipelinekonflikte in der DLX-Pipeline treten auf?
- 3. Die auftretenden Pipelinekonflikte sollen vom Compiler durch Einfügen von NOP-Befehlen behandelt werden. Ergänzen Sie das Programmstück so, dass die Pipelinekonflikte gelöst werden.
- 4. Welche NOP-Befehle sind noch erforderlich, wenn die Konflikte durch die HW erkannt und durch *Forwarding* behandelt werden?

Übung 6 – Aufgabe 2

Das folgende Programmstück soll auf einem Prozessor mit DLX-Pipeline ausgeführt werden:

```
R1, 1000 ; R1:= [1000]
S1: LOAD
S2: LOAD
          R2, 1004
                    ; R2:= [1004]
S3: ADD
          R3,R2,R1
                     R3 = R2 + R1
S4: ADD
          R1,R2,8
                     : R1 := R2 + 8
S5: SUB R4,R0,2
                     R4 = R0 - 2
S6: AND
          R5,R3,R2
                     ; R5:= R3 & R2
S7: STORE 1002,R4
                     ; [1002] := R4
                     ; [1006]:= R5
S8: STORE 1006,R5
S9: STORE 1008,R1
                     ; [1008]:= R1
```

Bestimmen Sie alle Datenabhängigkeiten im Programmstück

Übung 6 – Aufgabe 2

```
S1: LOAD
          R1, 1000
                    ; R1:= [1000]
S2: LOAD
          R2, 1004 ; R2:= [1004]
S3: ADD
          R3,R2,R1; R3:=R2+R1
S4: ADD R1,R2,8 ; R1:= R2 + 8
S5: SUB R4,R0,2 ; R4:= R0 - 2
S6: AND R5,R3,R2
                    : R5:= R3 & R2
S7: STORE 1002.R4
                     ; [1002]:= R4
S8: STORE 1006,R5
                     : [1006]:= R5
S9: STORE 1008,R1
                     ; [1008]:= R1
```

- Auftretende DLX-Pipelinekonflikte sollen durch Einfügen von NOP-Befehlen behoben werden. Ergänzen Sie das obige Programmstück so, dass es korrekte Ergebnisse liefert.
 - Sie sollen die Reihenfolge der Befehle nicht ändern und so wenig NOP-Befehle wie möglich einfügen.

Übung 6, Aufgabe 3

Das folgende Programmstück soll auf einem Prozessor mit DLX-Pipeline ausgeführt werden :

```
m1: ADD R1,R1,R1 ; R1:= R1 + R1
m2: ADD R2,R1,R1 ; R2:= R1 + R1
m3: ADD R2,R1,R2 ; R2:= R1 + R2
```

- Welchen Wert enthält das Register R2 nach Abarbeitung dieser Befehlsfolge, wenn R1 mit 4 und R2 mit 7 initialisiert ist. Die Pipelinekonflikte seien dabei nicht gelöst.
- 2. Ergänzen sie das Programm durch NOP-Befehle, so dass die Konflikte gelöst werden. Wie viele Takte benötigt das Programm?
- 3. Wie viele Takte benötigt das Programm bis zur vollständigen Leerung der Pipeline, wenn zusätzlich *Result Forwarding* eingesetzt wird?