

13.07.2006

Technische Grundlagen der Informatik II

10. Übung – MIPS Pipelining

Sommersemester 2006

Aufgabe 1: Taktdauer

a) Wie lange dauert jeweils die Ausführung der Befehle `load`, `store`, `add`, `beq` und `jump` bei der Mehrtakt-Implementierung mit einem variabel langem Takt (so kurz wie möglich für den aktuellen Befehl), wenn folgende Zeiten vorgegeben sind:

$$t_{regread} = 1 \text{ ns}$$

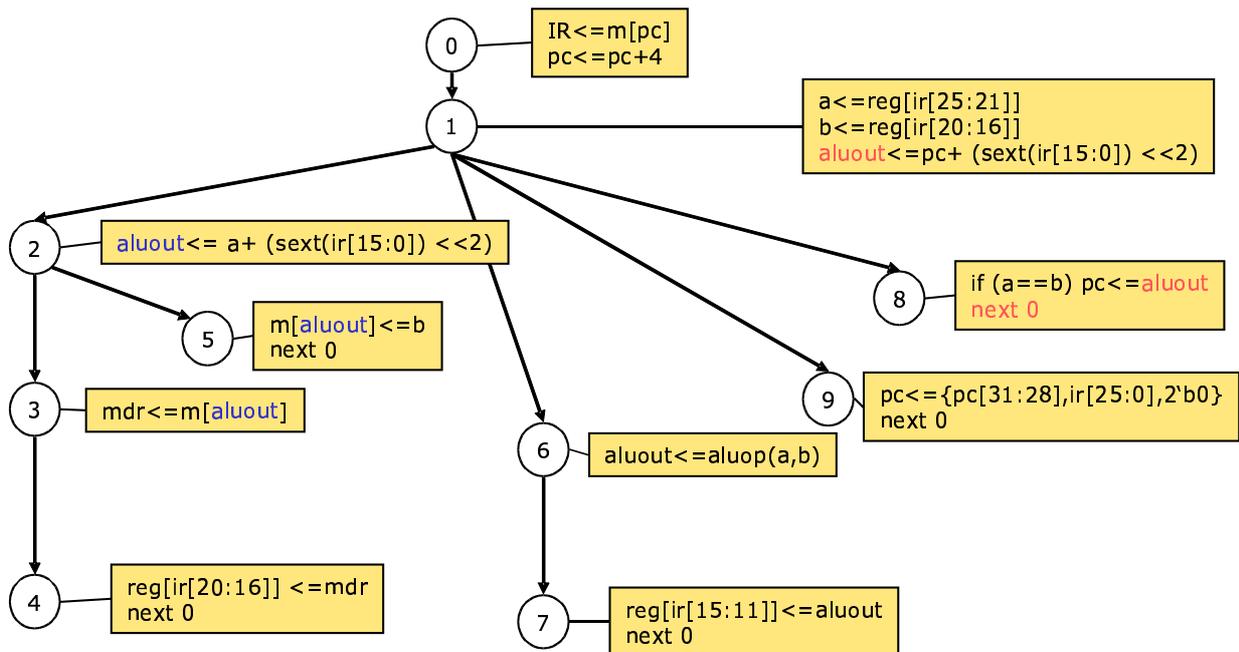
$$t_{regwrite} = 1 \text{ ns}$$

$$t_{ALU} = 2 \text{ ns}$$

$$t_{readmem} = 3 \text{ ns}$$

$$t_{writemem} = 3 \text{ ns}$$

Alle anderen Zeiten werden zu Null angenommen ($t_{cto} = 0$, $t_{setup} = 0$, $t_{mux} = 0$, $t_{and} = 0$, $t_{control} = 0$, usw.). Für jede Mikrooperation ist die minimale Taktperiode anzugeben und im folgenden Zustandsdiagramm einzutragen.



b) Geben Sie nun für jeden Befehl die Summe der Zeiten für die jeweils auszuführenden Mikrooperationen an.

c) Mit welchem Takt würden Sie eine MIPS-Pipeline bei den obigen Zeiten betreiben?

Aufgabe 2: Pipelining

Lassen Sie für eine MIPS-Pipeline ohne Forwarding die Befehlsfolge

```
add $2, $1, $3
sub $12, $2, $5
add $14, $2, $2
lw $15, 100($2)
```

durchlaufen, wobei die Register \$i am Anfang jeweils den Wert i besitzen. Schreiben Sie für jede einzelne Stufe (und je Befehl) in zeitlicher Abfolge auf, welche Werte für die nächste Stufe benötigt werden. Für einen Befehl lw \$6, 5(\$4) wäre dies z.B.:

```
IF ← lw $6, 5($4)
ID ← (lw, $6, 5, reg[4] = 4)
EX ← (lw, $6, 5 + 4 = 9)
MEM ← ($6, mem[9])
WB: reg[6] := mem[9]
```

Stellen Sie fest, welche Werte sich danach in den Zielregistern ergeben. Was müssen Sie ändern, um die gleichen Ergebnisse wie ohne Pipelining zu erhalten? Schreiben Sie dazu den geänderten Ablauf und die Ergebniswerte der Register auf.

Aufgabe 3: Pipeline in Verilog

Implementieren Sie die Berechnung von $y = (a + b) - (c + (d - 3))$

a) als kombinatorisches Schaltnetz

b) als 3-stufige Pipeline

in Verilog. Alle Werte sind 32 Bit breit.