

Exempel $F \rightarrow C$

C-kod:

```
float f2c(float f){
    return ((5.0/9.0)*(f-32));
}
```

MIPS:

```
f2c: lwc1 $f16, const5($gp)
     lwc1 $f18, const9($gp)
     div.s $f16, $f16, $f18
     lwc1 $f18, const32($gp)
     sub.s $f18, $f12, $f18
     mul.s $f $f $f
     jr    ra
```

Exempel från tenta

- a) Nyttjar lokaliteten bättre.
- b) För bara plats med 4 block \Rightarrow Fler konflikter.
- c) Vad är den genomsnittliga misskostnaden?

$$4B: 0.38(4 + 4u) = 1.9$$

$$16B: 0.22(4 + 16u) = 1.76$$

$$64B: 0.19(4 + 64u) = 3.8$$

$$256B: 0.27(4 + 256u) = 18.36$$

Exempel Hopp hazards

ADDI	\$5, \$0, 2	Always stall: 2 cykler extra pga hazards. Om hoppvillkor beräknas och används
L1: LW	\$4, 100(\$5)	i ID: $10 + 2 + 4 = 16$ (4: Töm pipe) Om Ex: $10 + 2 \cdot 3 + 4 = 20$
ADDI	\$5, \$5, -1	Assume not taken: Gissar fel en gång. \Rightarrow 1 cykel extra om hoppvillkoret beräknas
ADD	\$3, \$3, \$4	och används i ID-staget
BNE	\$5, \$0, L1	$10 + 1 + 4 = 15$
SW	\$3, 100, \$0	Om vi beräknar i Ex och använder i mem \Rightarrow 3 cykler extra.
		$10 + 3 + 4 = 17$

L1 körs 299.

Alla datahazards kan

lösas med forwarding.

Hur många extra cykler pga
hopp hazards?

Räkna inte med försörjda
hopp!

Multi-threading (MT)

Flera PC, fler registerfiler, snabb växling mellan trådar.

Fin resp grovkornig MT.

Parallelism exempel

Summera 10 tal sekventiellt och utför också en 10×10 matrisaddition. Vi kan använda 10 eller 100 CPUs

$$1 \text{ CPU: } T = (10 + 100) \cdot t_{\text{add}} = 110 \cdot t_{\text{add}}$$

$$100 \text{ CPU: } T = 10 \cdot t_{\text{add}} + \frac{100}{10} \cdot \dots$$