

App 1 hanya butuh 80% resources

App 2 hanya butuh 20% resources

App 1 memiliki fitur paralel sebanyak 40%

App 2 memiliki fitur paralel sebanyak 99%.

fraction enhanced App 1 = 0,4

fraction enhanced App 2 = 0,99

Jumlah prosesor = 2

$$\text{Speed up App 1} = \frac{1}{0,6 + \frac{0,4}{2}} = 1,25$$

$$\text{speed up App 2} = \frac{1}{0,01 + \frac{0,99}{2}} = 1,98$$

$$\begin{aligned}\text{speed up overall App 1} &= \frac{1}{\text{resource App 2} + \left(\text{resource App 1} \cdot \left[1 - \text{Fraction enhanced} + \frac{\text{Fraction}}{\text{Jumlah CPU}} \right] \right)} \\ &= \frac{1}{0,2 + 0,8 \cdot 0,8} = 1,19\end{aligned}$$

$$\begin{aligned}\text{speed up overall App 2} &= \frac{1}{\text{resource App 1} + \left(\text{resource App 2} \cdot \left[1 - \text{Fraction enhanced} + \frac{\text{Fraction enhanced}}{\text{Jumlah CPU}} \right] \right)} \\ &= \frac{1}{0,8 + 0,2 \cdot 0,805} = 1,109\end{aligned}$$

1,

Loop :

LD	F2, 0(Rx)	+3
MULTD	F2, F0, F2	+4
DIVD	F8, F2, F0	+10
LD	F4, 0(Ry)	+3
ADD	F4, F0, F4	+2
ADD	F10, F8, F2	+2
SD	F4, 0(Ry)	+1
ADDI	Rx, Rx, #8	+0
ADDI	Ry, Ry, #8	+0
SUB	R20, R4, Rx	+0
BNZ	R20, Loop.	+1

+

26 cycle, masing-masing instruksi dikenakan 1 cycle delay,

sehingga total cycle yang dibutuhkan oleh loop

per iterasi adalah $26 + 11 \text{ Instruksi} \cdot 1 \text{ cycle delay} = \underline{\underline{37 \text{ Cycle}}}$

~~LD F2, 0(Rx).~~

Pipeline 0

LD F2, 0(Rx)

<stall> LD

<stall> LD

<stall> LD

MULT F2, F0, F2

<stall> MULT

<stall> MULT

<stall> MULT

<stall> MULT

DIV F8, F2, F0

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

ADD F10, F8, F2

<stall karena ADD>

<stall karena ADD>

ADDI Rx, Ry, #8

<stall karena ADDI>

BNZ R20, Loop.

Pipeline 1

<nop>

<nop>

<nop>

<nop>

<nop>

<nop>

<nop>

<nop>

<nop>

LD(F4, 0(Ry)).

<stall karena LD>

<stall karena LD>

<stall karena LD>

ADD D #4, F0, F4.

<stall karena ADD>

<stall karena ADD>

<stall karena DIV>

<stall karena DIV>

<stall karena DIV>

SD F4, 0(Ry).

<stall karena SD>

ADDI Rx, Ry, #8

<stall karena ADDI>

SUB R20, R4, Rx

<nop>

Karena LD memakan 3 cycle + 1 cycle

Karena MULT memakan 4 cycle + 1 cycle

DIV membutuhkan 11 latency cycle.

LD membutuhkan 4 latency cycle.

26 cycle.

2.4. Jika instruksi $N+1$ dieksekusi di pipeline 1
di saat yang sama dengan instruksi N di pipeline 0
dan karena $N+1$ memiliki latency yang
lebih kecil dari ~~di~~ instruksi N , instruksi $N+1$
lebih dulu selesai. Berikut dua alasan
mengapa hal ini berbahaya!

1. Jika instruksi N dan $N+1$ menulis ke register yang
sama, maka perlu dipastikan bahwa instruksi
 N harus menulis lebih dulu dibandingkan instruksi
 $N+1$.
2. Jika instruksi $N+1$ adalah branch dan instruksi N ,
maka sebelum instruksi N selesai, instruksi $N+1$
tidak boleh menulis hasilnya ke register. Instruksi
 $N+1$ hanya bisa menulis hasil ke register jika konsisten
dengan branching dari instruksi N .

Contoh Instruksi $N+1$ yang dapat selesai lebih dulu
dari instruksi N adalah sbb.

Instruksi N : `DIVD F8, F2, F0`

Instruksi $N+1$: `LD F4, 0(R5)`

Instruksi $N+1$ akan selesai lebih dulu karena memiliki
execution latency yang lebih rendah dibandingkan
instruksi N . Execution delay `DIVD` lebih besar
dari execution delay `LD`.

Pipeline 0

Pipeline 1

LD F2, 0(Rx)
 <Stall karena LD>
 <Stall karena LD>
 <Stall karena LD>
 MULT F2, F0, F2
 <Stall karena MULT>
 <Stall karena MULT>
 <Stall karena MULT>
 <Stall karena MULT>
 DIVD F8, F2, F0
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 <Stall karena DIVD>
 ADD F10, F8, F2
 <Stall karena BNZ>

LD F4, 0(Ry)
 <Stall karena LD>
 <Stall karena LD>
 <Stall karena LD>
 ADD F4, F0, F4
 <Stall karena ADD>
 <Stall karena ADD>
 SD F4, 0(Ry)
 <Stall karena SD>
 ADDI Rx, Rx, #8
 ADDI Ry, Ry, #8
 SUB R20, R4, Rx
 <nop>
 <nop>
 <nop>
 <nop>
 <nop>
 <nop>
 <nop>
 <nop>
 <nop>
 <nop>
 BNZ R20, Loop

Jumlah Cycle = 22