CASPER

CS305: Computer Architecture

Delayed Branch and Branch Delay Slots

https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html

https://www.cse.iitb.ac.in/~biswa/

What else can be done? Compiler?

Delayed branch: Define branch to take place AFTER a following instruction(used to be in early RISC processors)

branch instruction

sequential successor$_1$

sequential successor$_2$

....

sequential successor$_n$

branch target if taken

Branch delay of length *n*

One slot: Not Taken (untaken) branch

branch instruction

sequential successor$_1$

branch target if taken

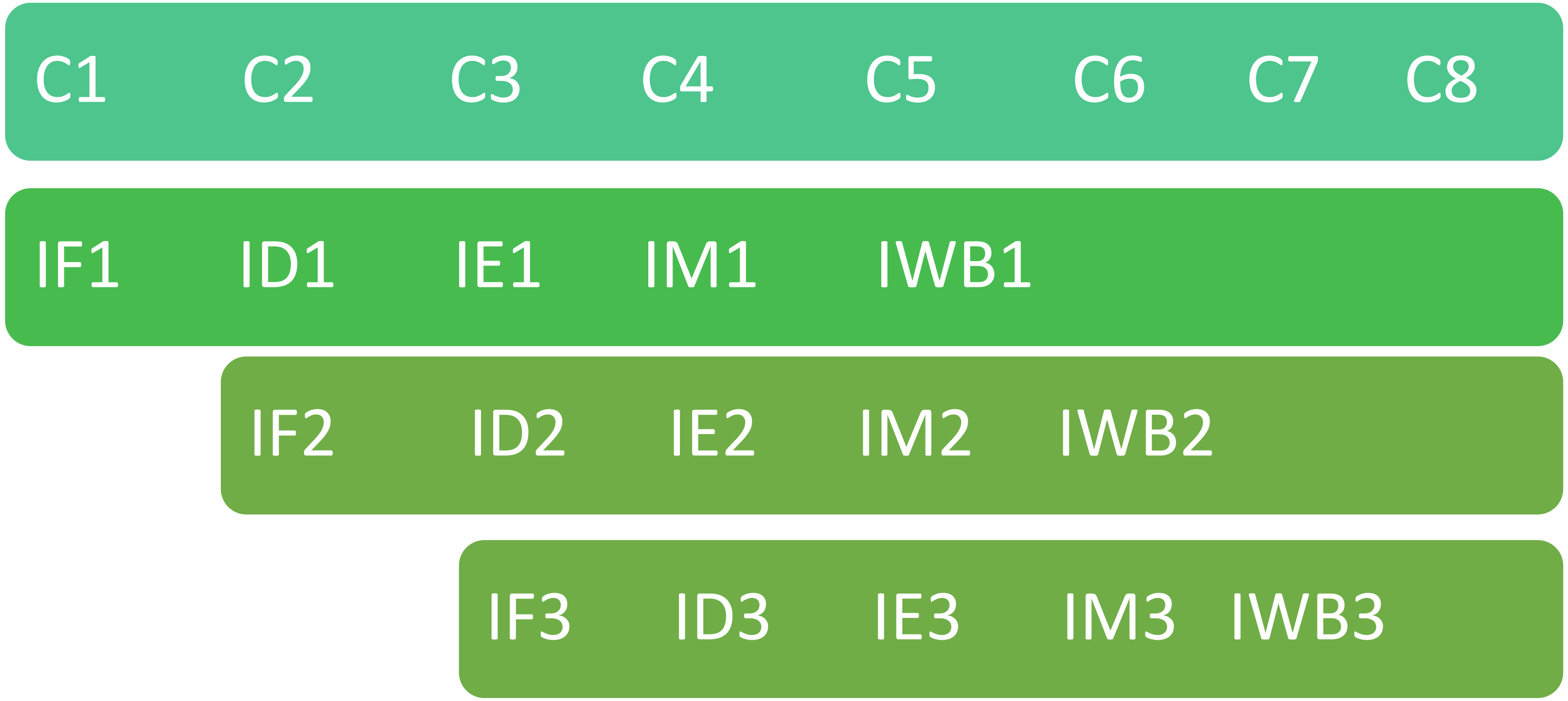
	C1	C2	C3	C4	C5	C6	C7	C8
Untaken branch inst. (i)	IF1	ID1	IE1	IM1	IWB1			
Branch delay inst. (i+1)		IF2	ID2	IE2	IM2	IWB2		
Instruction i+2			IF3	ID3	IE3	IM3	IWB3	

One slot: Taken branch

branch instruction

sequential successor$_1$

branch target if taken

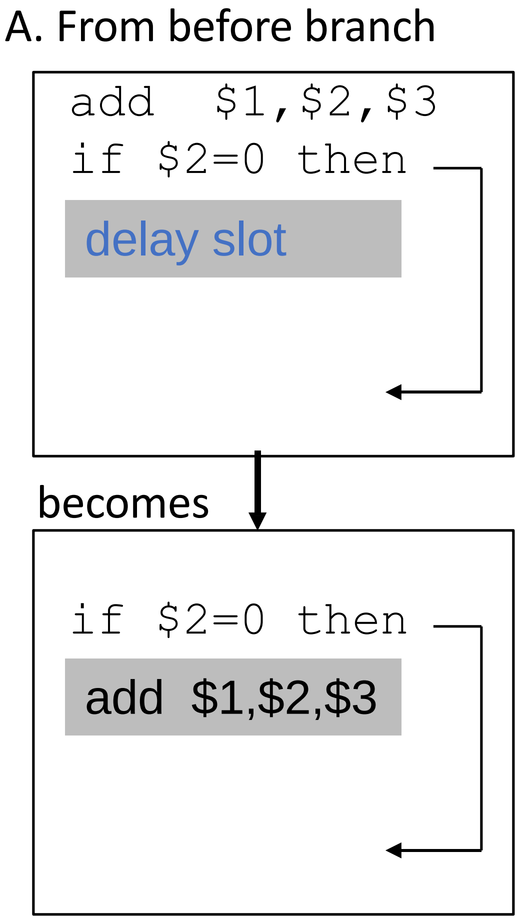


	C1	C2	C3	C4	C5	C6	C7	C8
Taken branch inst. (i)	IF1	ID1	IE1	IM1	IWB1			
Branch delay inst. (i+1)		IF2	ID2	IE2	IM2	IWB2		
Branch Target			IF3	ID3	IE3	IM3	IWB3	

Word of Caution!

Do not put a branch
in the branch delay slot ☹

Scheduling Branch Delay Slots



A. From before branch

```
add  $1,$2,$3
if $2=0 then
```

delay slot

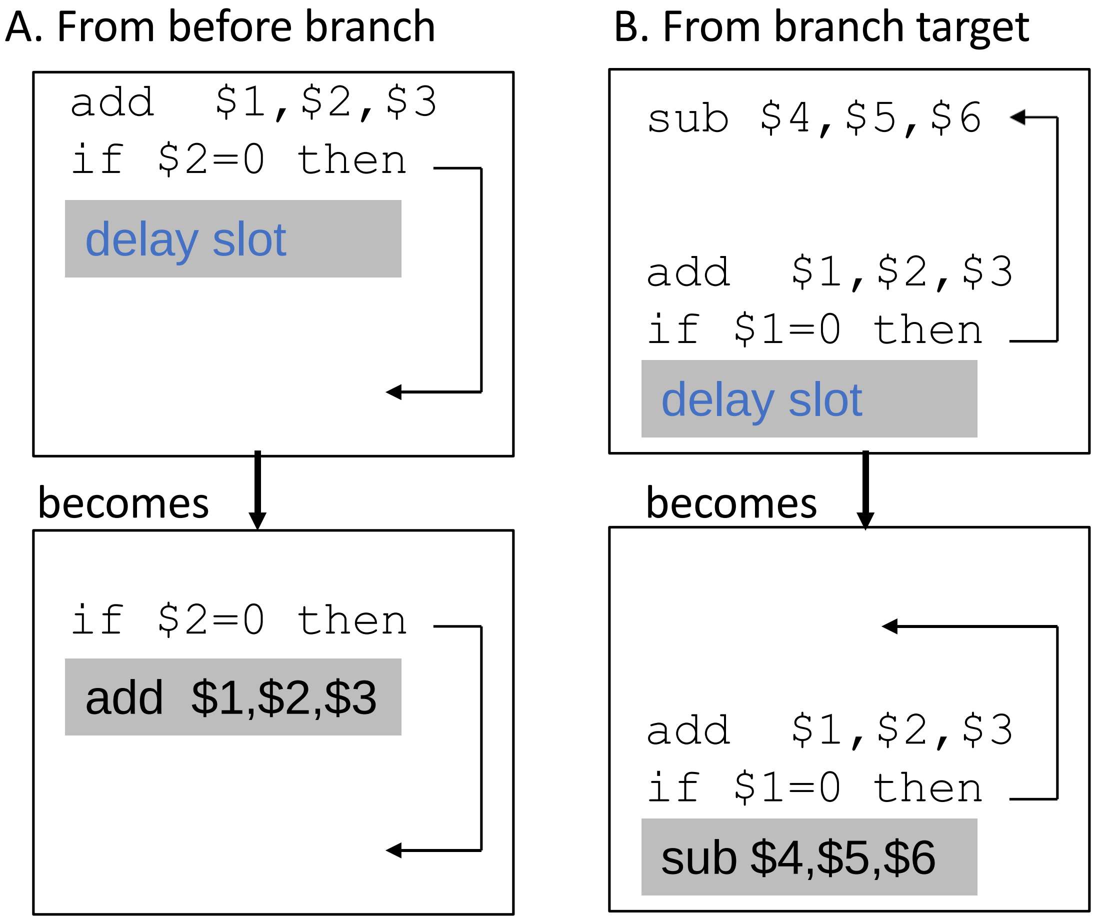
becomes

```
if $2=0 then
```

add $1,$2,$3

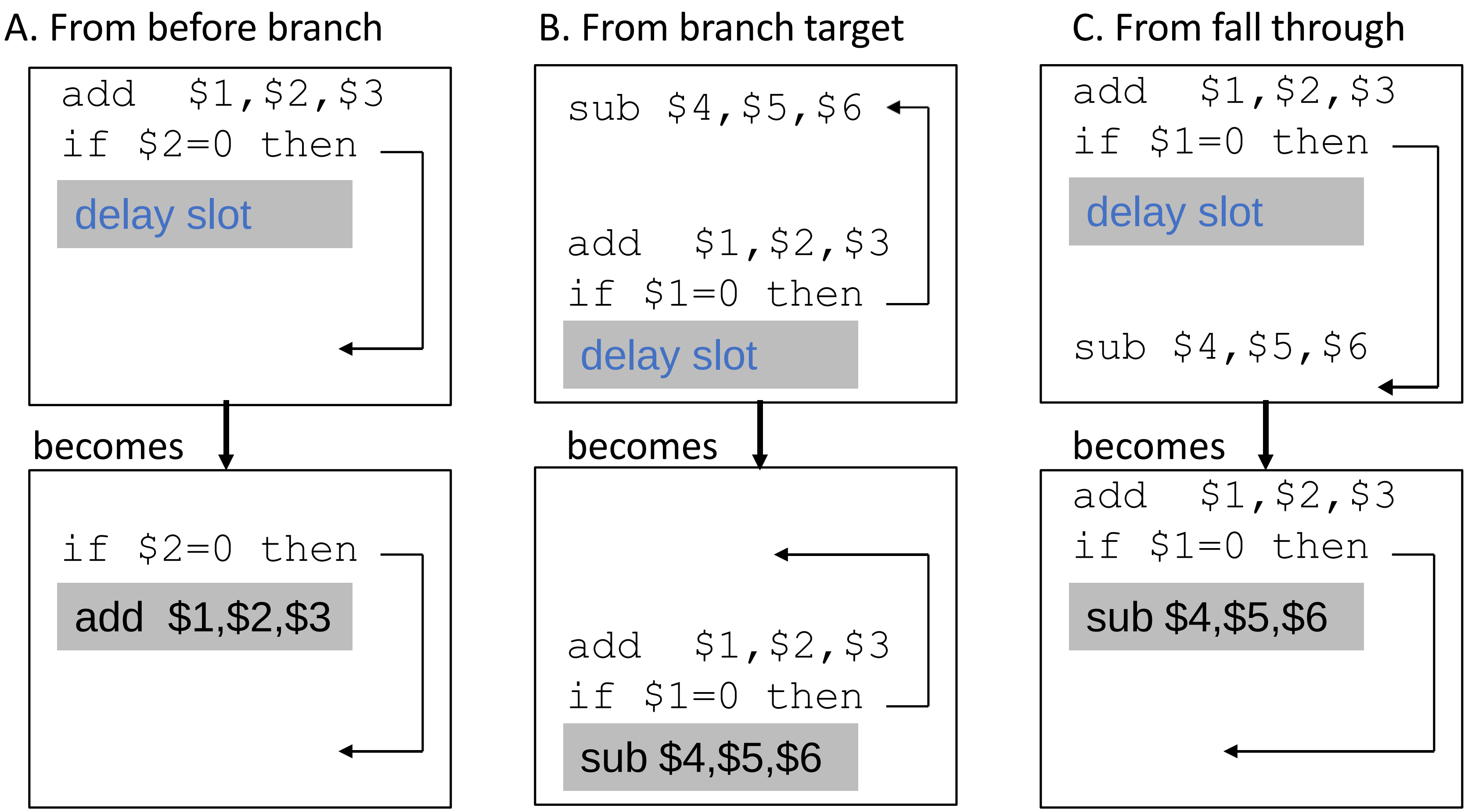
A is the best choice, fills delay slot & reduces instruction count (IC)

Scheduling Branch Delay Slots



A is the best choice, fills delay slot & reduces instruction count (IC)

Scheduling Branch Delay Slots



A. From before branch

```
add  $1,$2,$3
if $2=0 then
delay slot
```

becomes

```
if $2=0 then
add  $1,$2,$3
```

B. From branch target

```
sub $4,$5,$6


add  $1,$2,$3
if $1=0 then
delay slot
```

becomes

```
add  $1,$2,$3
if $1=0 then
sub $4,$5,$6
```

C. From fall through

```
add  $1,$2,$3
if $1=0 then
delay slot

sub $4,$5,$6
```

becomes

```
add  $1,$2,$3
if $1=0 then
sub $4,$5,$6
```

A is the best choice

Stalls and Performance

For a program with N instructions and S stall cycles,

$$\text{Average CPI} = \frac{N}{N}$$

Stalls and Performance

For a program with N instructions and S stall cycles,

$$\text{Average CPI} = \frac{N+S}{N}$$

New Pipeline Speedup

$$\text{Pipeline Speedup} = \frac{\text{Pipeline Depth}}{1+\text{pipeline stalls because of branches}}$$

Pipeline stalls (branches) = Branch frequency X penalty

Summary

Data Hazards

Bypassing/forwarding

Stalls (NOPs) – if no scope for bypassing

Control hazards

Speculate, PC=PC+4, kill the wrong path

Delayed branch with the help of branch delay slots, new pipeline speedup

Takk