

# 计算机体系结构

周学海 xhzhou@ustc.edu.cn 0551-63492149 中国科学技术大学



## Review

## · 指令级并行(ILP):流水线的平均CPI

- Pipeline CPI = Ideal Pipeline CPI + Struct Stalls + RAW Stalls+ WAR Stalls + WAW Stalls + Control Stalls + .....
- 提高指令级并行的方法
  - 软件方法: 指令流调度,循环展开,软件流水线, trace scheduling
  - 硬件方法

## ・ 软件方法: 指令流调度-循环展开

- 指令调度, 必须保证程序运行的结果不变
- 寄存器的重命名
- 循环步长的调整
- 偏移量的修改
- 保证存储器访问无冲突



## 第5章 指令级并行

## 5.1 指令级并行的基本概念及静态指令流调度

ILP及挑战性问题 软件方法挖掘指令集并行

### 5.2硬件方法挖掘指令级并行

- 5.2-1 指令流动态调度方法之一: Scoreboard
- 5.2-2 指令流动态调度方法之二: Tomasulo
- 5.3 分支预测方法
- 5.4 基于硬件的推测执行
- 5.5 存储器访问冲突消解及多发射技术
- 5.6 多线程技术



# 5.2-1 指令流动态调度方法: Scoreboard

记分牌 技术要点

示例



## 硬件方案: 指令级并行

- · 为什么要使用硬件调度方案?
  - 在编译时无法确定的相关,可以通过硬件调度来优化
  - 编译器简单
  - 代码在不同组织结构的机器上,同样可以有效的运行
- · 基本思想: 允许 stall后的指令继续向前流动

DIVD F0,F2,F4

**ADDD F10,F0,F8** 

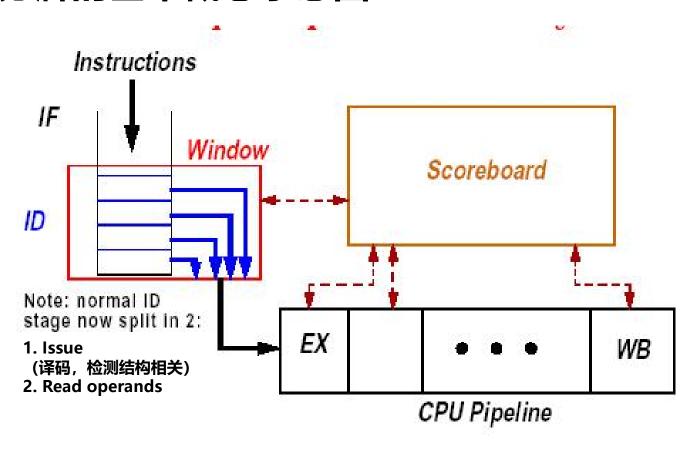
**SUBD F12,F8,F14** 

一允许乱序执行(out-of-order execution) => 乱序完成(out-of-order completion)



## 硬件方案之一: 记分牌

## ・记分牌的基本概念示意图



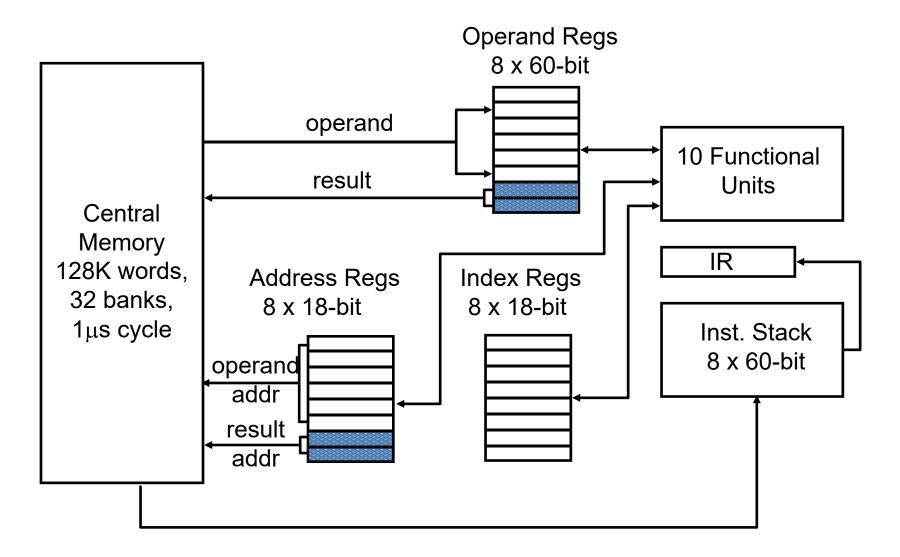


## 记分牌技术要点(1/2)

- · Scordboard技术将ID 段分为:
  - Issue—译码, 检测结构相关
  - Read operands—等待到无数据相关时,读操作数
- · 起源于1964年Control Data Corporation推出的CDC6600: 顺序发射,乱序 执行,乱序完成,没有采用定向技术,只实现非精确中断
  - Ten Functional Unit
    - Floating ADD, Floating Multiply(2), Floating Divide
    - Fix ADD, Increment (2), Boolean, Shift, Branch
  - Ten Peripheral Processors for Input/Output: a fast time-shared 12-bit integer ALU
  - Load /store结构
- ・ 集中相关 (冲突) 检查(<mark>Scoreboard</mark>), 互锁机制 (interlock) 解决相关
  - 一种装置,其中一个部分或机构的运转自动地导致或阻止另一个部分的运转
- · 采用这种技术的微处理器企业
  - MIPS, HP, IBM
  - Sun 公司的UltraSparc
  - DEC Alpha

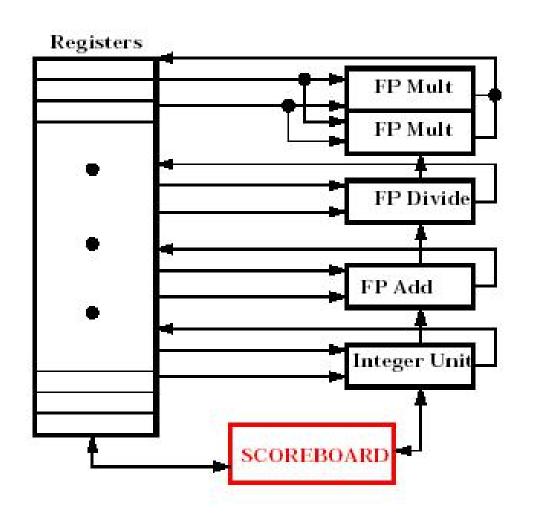


## CDC 6600: Datapath





## 带有记分牌控制的MIPS



Note: this model could support both single or multi-issue

Exception is that one multiply will be issued per cycle

All depends on bus/trunk structure



## 记分牌技术要点(2/2)

- Out-of-order completion => WAR, WAW hazards?
- WAR: 一般解决方案 (在不采用寄存器重命名的情况下)
  - 对操作排队
  - 仅在读操作数阶段读寄存器
- · WAW: 检测到相关后,停止发射前一条指令,直到前一条指令完成
- · 提高效率的前提: 需要有多条指令进入执行阶段=>必须有 多个执行部件或执行部件是流水化的
- · 记分牌保存相关操作和状态
- ・ 指令执行过程: IF, ISSUE, RO, EX, WR



## 记分牌控制的四阶段(1/2)

### · 1.Issue—指令译码,检测结构相关

- 准入条件: 顺序发射, 并且没有结构相关, 没有WAW相关
- 如果当前指令所使用的功能部件空闲,并且没有其他活动的指令使用相同的目的寄存器 (WAW),记分牌发射该指令到功能部件,并更新记分牌内部数据,如果有结构相关或WAW相关,则该指令的发射暂停,并且也不发射后继指令,直到相关解除。

### · 2. Read operands—没有数据相关时,读操作数

- 准入条件: 没有RAW相关。
- 如果先前已发射的正在运行的指令不对当前指令的源操作数寄存器进行写操作,或者一个正在工作的功能部件已经完成了对该寄存器的写操作,则该操作数有效。操作数有效时,记分牌控制功能部件读操作数,准备执行。
- 记分牌在这一步动态地解决了RAW相关,指令可能会乱序执行。



## 记分牌控制的四阶段 (2/2)

- · 3.Execution—取到操作数后执行 (EX)
  - **准入条件**: RO后准入
  - 接收到操作数后, 功能部件开始执行
  - 当计算出结果后,通知记分牌,可以结束该条指令的执行。
- 4.Write result—finish execution (WR)
  - 准入条件: 没有WAR相关
  - 一旦记分牌得到功能部件执行完毕的信息后,记分牌检测WAR相关, 如果没有WAR相关,就写结果,如果有WAR相关,则暂停该条指令。
  - Example:

DIVD F0,F2,F4

ADDD F10,F0,F8

SUBD F8,F8,F14

CDC 6600 scoreboard 将暂停 SUBD 直到ADDD 读取操作数后,才进入WR段处理。



## 记分牌的结构

- 1.Instruction status—记录正在执行的各条指令的状态步
- 2.Functional unit status—记录功能部件(FU)的状态。用9个域记录每个功能部件的9个参量:

Busy—指示该部件是否空闲

Op—该部件所完成的操作

Fi—其目的寄存器编号

Fj, Fk—源寄存器编号

Qj, Qk—产生源操作数Fj, Fk的功能部件

Rj, Rk—标识源操作数Fj, Fk是否就绪的标志

3.Register result status—如果存在功能部件对某一寄存器进行写操作, 在寄存器结果状态表中记录该功能部件。如果没有指令对该寄存器进行 写操作,则该域为Blank

 寄存器索引
 FO
 F2
 F4
 F6
 ......

 部件编号
 Image: Control of the property of th



# 记分牌流水线控制

Instruction status	Wait until	Bookkeeping
Issue	Not busy (FU) and not Result(D)	Busy(FU)← yes; Op(FU)← op; Fi(FU)← `D'; Fj(FU)← `S1'; Fk(FU)← `S2'; Qj← Result('S1'); Qk← Result(`S2'); Rj← not Qj; Rk← not Qk; Result('D')← FU;
Read operands	Rj and Rk	Rj← No; Rk← No
Execution complete	Functional unit done	
Write result	∀f((Fj( f )≠Fi(FU) or Rj( f )=No) & (Fk( f ) ≠Fi(FU) or Rk( f )=No))	∀f(if Qj(f)=FU then Rj(f)← Yes); ∀f(if Qk(f)=FU then Rk(f)← Yes); Result(Fi(FU))← 0; Busy(FU)← No

Result(): 寄存器结果状态表, FU: 功能部件



# 5.2-1 指令流动态调度方法: Scoreboard

记分牌 技术要点

示例

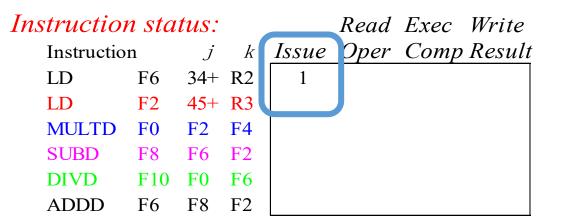
2022-4-12 xhzhou@USTC 15



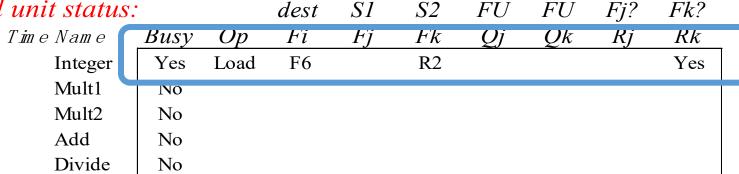
# Scoreboard Example

LD F6 34+ R2 LD F2 45+ R3 MULTD F0 F2 F4 SUBD F8 F6 F2			Issue	Read operand		iWrite tResult		Mı Ac	FU ad - 1/ ulti - 10	)个Cycle 个Cycle	
DIVD F10	F0	F6									
ADDD F6	F8	F2									
<u>Functional</u>	unit	status	_		dest	S1	<i>S2</i>	FU for	FU for	kFj?	Fk?
Time	Name	<del>)</del>	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Rj	Rk
	Inte	ger	No								
	Mult	1	No								
	Mult	2	No								
	Add		No								
	Divi	de	No								
<u>Register re</u>	sult	status	_								
Clock			F0	F2	F4	<i>F6</i>	F8	<i>F10</i>	F12		F30
		FU									



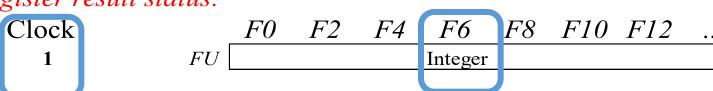


#### Functional unit status:

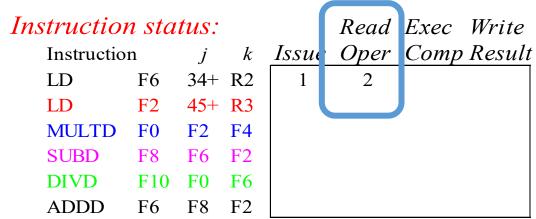


*F30* 

#### Register result status:







#### Functional unit status:

Time Name
Integer
Mult1
Mult2
Add
Divide

		dest	SI	<i>S2</i>	FU	FU	FJ?	FK?
Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Yes	Load	F6		R2				Yes
No								
No								
No								
No								

#### Register result status:

Clock

FU

<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10	<i>F12</i>	• • •	F30

Integer

Issue 2nd LD?



```
Instruction status:
                                    Exec
                                           Write
                                    Comp Result
   Instruction
                 j k
                        Issue Oper
                34+ R2
   LD
           F6
                                       3
           F2
                45+ R3
   LD
   MULTD
           F0
                F2
                    F4
   SUBD
           F8
   DIVD
           F10
                    F6
                F0
   ADDD
                F8
                    F2
           F6
```

#### Functional unit status:

Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F6		R2				No
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

S1 S2 FU FU

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30 FU Integer

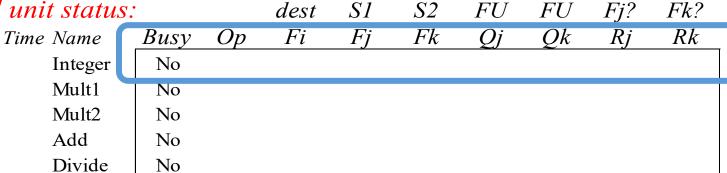
dest

Issue MULT?



```
Read Exec
Instruction status:
                                               Write
                           Issue Oper Comp Result
   Instruction
                        k
   LD
                  34 + R2
                                    2
                                           3
             F6
                              1
                                                  4
   LD
             F2
                  45+ R3
   MULTD
             F0
                  F2
                      F4
   SUBD
             F8
                  F6
                      F2
   DIVD
             F10
                      F6
                  F<sub>0</sub>
                  F8
   ADDD
             F6
                      F2
```

#### Functional unit status:



#### Register result status:





Instruction	n sta	tus:			Read	Exec	Write
Instruction	Instruction		k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5			
MULTD	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

#### Functional unit status:

	<i>.</i>		acsi	$\mathcal{O}_{\mathbf{I}}$	52	10	10	<b>1</b> J.	I IV.	
Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Rj	Rk	
Integer	Yes	Load	F2		R3				Yes	
Mult1	No									
Mult2	No									
Add	No									
Divide	No									

SI

S2

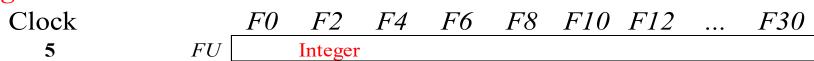
FU

FII

Fi?

Fk?

#### Register result status:

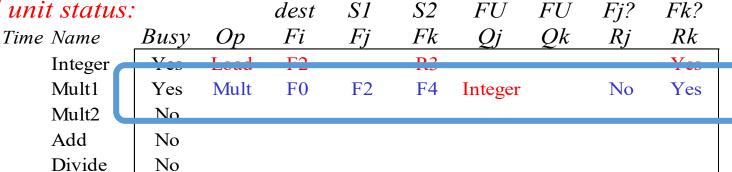


dest



Instruction	n sta	tus:			Read	Exec	Write
Instruction	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6		
MULTD	F0	F2	F4	6			
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

#### Functional unit status:



#### Register result status:

Clock	$\_F0$	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10	<i>F12</i>	• • •	F30
6	FU Mult1	Integer							



Instructio	on sta	tus:			Read	Exec	Write
Instructi	on	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
M ULTD	FO	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

#### Functional unit status: SI *S2* FUFk? dest FUFj? FiBusv FiFkOi Qk RiRkTime Name Op F2 **R3** No Yes Load Integer Mult1 Mult Yes F<sub>0</sub> F2 F4 Integer Yes No Mult2 Add Yes F8 F2 Sub F6 Integer Yes No Divide $N_{\Delta}$

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30 FU Mult1 Integer Add

Read multiply operands?



# Scoreboard Example: Cycle 8a (First half of clock cycle)

Instruction	n sta	tus:			Read	Exec	Write
Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
MULTD	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

#### Functional unit status:

				, <del>-</del> -				- ) ·	
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F2		R3				No
Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2	I	nteger	Yes	No
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

SI

*S2* 

FU

FU

Fi?

Fk?

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F308 FU Mult1 Integer Add Divide

dest

Issue DIVD



# Scoreboard Example: Cycle 8b (Second half of clock cycle)

Instruction			Read	Exec	Write		
Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional unit status:			dest	S1	<i>S2</i>	FU	FU	Fj?	Fk?
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2			Yes	Yes
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30

8 FU Mult1 Add Divide

2nd LD 写结果,并通知相关指令结果可用,释放IU部件



C1

- 1					A
	m ctv	רדי זו וי	$\alpha$	CTA	<i>T1  </i> C '
	nstr	$\mu \cup \iota \iota$	$\boldsymbol{O}$	DLU	$\iota u \circ \iota$
			• • •	~ • • • •	

Instruction	n	j	k
LD	F6	34+	R2
LD	F2	45+	R 3
<b>MULTD</b>	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	$\mathbf{F0}$	F6
ADDD	F6	F8	F2

Read	Exec	Write
110000	11100	,,,,,,,

Issue	Oper	Comp	Result
1	2	3	4
5	6	7	8
6	9		
7	9		
8			

#### Functional unit status:

	1
Note	-
Remaining	

Time Name
Integer
10 Mult1
Mult2

2 Add Divide

۶.	-		aest	$\mathcal{S}I$	32	FU	FU	FJ?	PK!	
	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	
	No									Ĭ
	Yes	Mult	F0	F2	F4			Yes	Yes	
	No									
	Yes	Sub	F8	F6	F2			Yes	Yes	
	Yes	Div	F10	F0	F6	Mult1		No	Yes	

CO

 $\mathbf{L}\mathbf{I}\mathbf{I}$ 

 $\mathbf{L}^{T}$ 

 $\Gamma$ :2

 $\Gamma I_{-} 2$ 

#### Register result status:

Clock 9

FU

F0	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	<i>F10</i>	F12	•••	F30
Mult1				Add	Divide			

## · Read operands for MULT & SUB? Issue ADDD?



7					A
_/	nstr	11 <i>CT</i> 1	on	$\mathbf{C}TA$	7776.
_		$u \cup u$	OII	DIU	

Instruction	n	j	k
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Read	Exec	Write
110000	Livee	,,,,,,,

$\underline{I}$	ssue	Oper	Comp	Result
	1	2	3	4
	5	6	7	8
	6	9		
	7	9		
	8			

#### Functional unit status:

Time	Name
	Integer
9	Mult1
	Mult2
1	Add
	Divide

:		dest	SI	<i>S2</i>	FU	FU	Fj?	Fk?
Busy	v Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
No								
Yes	Mult	F0	F2	F4			No	No
No								
Yes	Sub	F8	F6	F2			No	No
Yes	Div	F10	F0	F6	Mult1		No	Yes

#### Register result status:

Clock 10

F0	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10	F12	•••	
Mult1				Add	Divide			

*F30* 



C1

7		1		, •	1 1	
	nc	TV	11/1	100	statu	$\mathbf{c}$ .
		$\iota$	$n \cup \iota$	$\iota \cup \iota \iota$	simin	<b>.</b>

Instruction	j	k	
LD	F6	34+	R2
LD	F2	45+	R3
MULTD	F0	F2	F4
SUBD	F8	F6	F2
DIVD	F10	F0	F6
ADDD	F6	F8	F2

Issu	e Oper	Comp	Result
1	2	3	4
5	6	7	8
6	9		
7	9	11	
8			

#### Functional unit status:

Time	Name
	Integer
8	Mult1
	Mult2
0	Add
	Divide

FU

•			aest	$\mathcal{S}I$	32	FU	FU	FJ?	FK?
	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
	No								
	Yes	Mult	F0	F2	F4			No	No
	No								
	Yes	Sub	F8	F6	F2			No	No
	Yes	Div	F10	F0	F6	Mult1		No	Yes

 $C_{2}$ 

 $\mathbf{L}\mathbf{I}\mathbf{I}$ 

 $\mathbf{L}\mathbf{I}\mathbf{I}$ 

 $\Gamma$ :2

 $\Gamma I_{-} 2$ 

*F30* 

#### Register result status:

Clock 11 
 F0
 F2
 F4
 F6
 F8
 F10
 F12
 ...

 Mult1
 Add Divide

2022-4-12 xhzhou@USTC 28



Instruction	n sta	tus:		Read	Exec	Write	
Instruction		j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

#### Functional unit status:

i dirett bedeedib.			CCSI	$\mathcal{O}_{\mathbf{I}}$	02	10	1 0	<b>1</b> J.	I IV.	
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	
Integer	No									
7 Mult1	Yes	Mult	F0	F2	F4			No	No	
Mult2	No									
Add	No									
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes	ĺ

SI

S2

FU FU

Fi?

Fk?

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30 12 FU Mult1 Divide

dest

### Read operands for DIVD?



Instruction	ı sta	tus:		Read	Exec	Write	
Instruction	Instruction		k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13			

#### Functional unit status:

i mill sialus.			uesi	$\mathcal{O}I$	02	I	I	IJ:	I IV.	
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	
Integer	No									
6 Mult1	Yes	Mult	F0	F2	F4			No	No	
Mult2	No									
Add	Yes	Add	F6	F8	F2			Yes	Yes	
Divide	Yes	Div	F10	$\mathbf{F0}$	F6	Mult1		No	Yes	

51

S2

FII

FII

Fi2

Fk2

#### Register result status:

Clock	F0	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10 F1.	2	F30
13	FU Mult1			Add		Divide		

dest

2022-4-12 xhzhou@USTC 30



C1

Instruction	n sta	tus:		Read	Exec	Write	
Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14		

<b>—</b> . • 1	• , , ,
Hunctional	' unit status:
Tancuona	milli sialas.

	~
<sup>r</sup> ime	Name
	Integer
5	Mult1
	Mult2
2	Add
	Divide

•		aest	$\mathcal{S}I$	32	FU	FU	FJ?	FK?
Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
No								
Yes	Mult	F0	F2	F4			No	No
No								
Yes	Add	F6	F8	F2			Yes	Yes
Yes	Div	F10	F0	F6	Mult1		No	Yes

 $C_{2}$ 

 $\mathbf{L}\mathbf{I}\mathbf{I}$ 

 $\mathbf{L}\mathbf{I}\mathbf{I}$ 

 $\Gamma$ :2

L1-2

#### Register result status:

Clock 14 F0 F2 F4 F6 F8 F10 F12 ... F30 FU Mult1 Add Divide



Instruction	n sta	tus:		Read	Exec	Write	
Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
$\Delta$ DDD	F6	F۶	F2	13	14		

#### Functional unit status:

ime	Name
	Integer
4	Mult1
	Mult2
1	Add
	Divide

		dest	SI	<i>S2</i>	FU	FU	FJ!	PK!
Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
No								
Yes	Mult	F0	F2	F4			No	No
No								
Yes	Add	F6	F8	F2			No	No
Yes	Div	F10	F0	F6	Mult1		No	Yes

771-9

#### Register result status:

Clock 15



Write

11	<i>istruction</i>	n sta	tus:			Read	Exec
	Instructio	n	j	k	Issue	Oper	Com
	I D	П.	24.	<b>D</b> 2	4	2	2

Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14	16	

#### Functional unit status: dest 51

i unii siaius.			uesi	$\mathcal{D}I$	02	I	I	IJ:	I N:	
Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Rj	Rk	_
Integer	No									
3 Mult1	Yes	Mult	F0	F2	F4			No	No	
Mult2	No									
0 Add	Yes	Add	F6	F8	F2			No	No	
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes	

52

FII

FII

Fi?

Fk?

#### Register result status:

Clock	$_{F0}$	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10 F12	•••	F30
16	FU Mult1			Add		Divide		

2022-4-12 xhzhou@USTC 33



**DIVD** 

**ADDD** 

# Scoreboard Example: Cycle 17

Instr	uction	ı sta	tus:			Read	Exec	Write	
In	struction	1	j	k	Issue	Oper	Comp	Result	
L	D	F6	34+	R2	1	2	3	4	
L	D	F2	45+	R3	5	6	7	8	
M	IULTD	F0	F2	F4	6	9			
SI	UBD	F8	F6	F2	7	9	11	12	

8

13

14

WAR Hazard!

#### Functional unit status: *S2* SI dest FUFUFj? Fk? Fi $F_i$ Qk Busy Fk $R_i$ RkTime Name Op No Integer 2 Mult1 F4 Yes Mult F0 No No Mult2 No F8 Add Yes Add F6 F2 No No Mult1 Divide F<sub>0</sub> F6 Div Yes F10 No Yes

16

#### Register result status:

F10

F6

F<sub>0</sub>

F8

F6

F2

Clock F0 F2 F4 F6 F8 F10 F12 ... F30  $\overline{F}$  FU Mult1 Add Divide

Why not write result of ADD???



Exec Write

Instruction s	status:			Read
Instruction	j	k	Issue	Oper

Instructio	n	j	$\boldsymbol{k}$	<i>Issue</i>	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
<b>MULTD</b>	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14	16	

#### Functional unit status:

ime	Name
	Integer
1	Mult1
	Mult2
	Add
	Divide

•			aest	$\mathcal{S}I$	32	FU	FU	FJ?	FK?
	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
	No								
	Yes	Mult	F0	F2	F4			No	No
	No								
	Yes	Add	F6	F8	F2			No	No
	Yes	Div	F10	F0	F6	Mult1		No	Yes

 $\Gamma$ :2

 $\Gamma I_{-} 2$ 

#### Register result status:

Clock 18

FU

F0	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10 F12	•••	F30
Mult1			Add		Divide		



Instruction	n sta	tus:		Read	Exec	Write	
Instruction	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14	16	

#### Functional unit status:

i mill sialus.			uesi	$\mathcal{O}I$	02	I	I	IJ:	I IV:	
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	
Integer	No									
0 Mult1	Yes	Mult	F0	F2	F4			No	No	
Mult2	No									
Add	Yes	Add	F6	F8	F2			No	No	
Divide	Yes	Div	F10	$\mathbf{F0}$	<b>F6</b>	Mult1		No	Yes	

S2

FII

FII

Fi2

Fk2

51

#### Register result status:

Clock	$\_F0$	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10	F12	•••	F30
19	FU Mult1			Add		Divide			

dest

2022-4-12 xhzhou@USTC 36



Instruction	n sta	tus:		Read	Exec	Write	
Instruction	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14	16	

#### Functional unit status:

correct Secretion.	•		CICSI	$\mathcal{O}_{\mathbf{I}}$	22	1 0	1 0	<b>1</b> J.	ı ıv.	
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	_
Integer	No									
Mult1	No									
Mult2	No									
Add	Yes	Add	F6	F8	F2			No	No	
Divide	Yes	Div	F10	F0	<b>F6</b>			Yes	Yes	

FU

FU

Fi?

Fk?

#### Register result status:

Clock		F0	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10 F	712	•••	<i>F30</i>
20	FU				Add		Divide			

dest



Instruction	n sta	tus:			Read	Exec	Write
Instruction	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21		
ADDD	F6	F8	F2	13	14	16	

Yes

Yes

Add

Div

Functional unit status		aest	SI	<i>S2</i>	FU	FU	FJ!	FK!	
Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	No								
Mult2	No								

F6

F10

F8

F<sub>0</sub>

F2

F6

No

Yes

No

Yes

#### Register result status:

Laurational and atatasa

Clock F0 F2 F4 F6 F8 F10 F12 ... F30 **21** FU Add Divide

· WAR Hazard is now gone...

Add

Divide



Instruction	n sta	tus:			Read	Exec	Write
Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21		
ADDD	F6	F8	F2	13	14	16	22

<b>7</b> 7 ,• 1	• ,	4 4
<b>Functional</b>	111111	ctatuc.
Tuncuona	uni	siains.

	'		00000	~ -	~ -			<i>- j</i> ·	<b>1</b>	
Time Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	_
Integer	No									
Mult1	No									
Mult2	No									
Add	No									
39 Divide	Yes	Div	F10	F0	F6			No	No	

FU FU Fi? Fk?

#### Register result status:

Clock	$\_F0$	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10 F12	•••	F30
22	FU					Divide		

dest

2022-4-12 xhzhou@USTC 39



## Continue.....



Instruction	n sta	tus:			Read	Exec	Write
Instruction	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	<b>F4</b>	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	
ADDD	F6	F8	F2	13	14	16	22

#### Functional unit status:

t tille steetis.	1		CCSI	$\mathcal{O}_{\mathbf{I}}$	02	1 0	1 0	<b>1</b> J.	I IV.	
Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Rj	Rk	_
Integer	No									
Mult1	No									
Mult2	No									
Add	No									
0 Divide	Yes	Div	F10	F0	F6			No	No	

S2

Fk?

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30 **61** FU Divide

dest



Instruction	n sta	tus:			Read	Exec	Write
Instructio	n	j	k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

#### Functional unit status:

Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Řj	Rk
Integer	No								
Integer Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Fj?

FU FU

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30
62 FU

dest

2022-4-12 xhzhou@USTC 42



## Review: Scoreboard Example: Cycle 62

Instruction status:					Read	Exec	Write
Instructio	Instruction		k	Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

<b>Functional</b>	'unit	status.
Tuncuonai	unii	siaius.

Time Name	Busy	Ор	Fi	Fj	Fk	Qj	Qk	Řj	Rk
Integer	No								
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

S1

S2

FU

FU

Fi?

Fk?

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30
62 FU

dest

- ・ 顺序issue; 乱序 execute & 乱序 commit
- · 问题: Branch指令怎么办?

2022-4-12 xhzhou@USTC 43



## 为什么顺序发射?

- · 顺序发射使我们可以进行程序的数据流分析
  - 我们可以知道某条指令的结果会流向哪些指令
  - 如果我们乱序发射,可能会混淆RAW和WAR相关
- · 每一周期发射多条指令也使用该原则将会正确地 工作
  - 寄存器文件至少需要有2x 个读端口和x个写端口.
  - 当有寄存器重命名时,例如: Tomasulo 还需要
  - 多端口的 "rename table" ,以同时对一组指令所用的寄存器重命名
  - 在单周期内发射到多个RS中



## Summary

## 硬件方法挖掘ILP

- 编译阶段无法确定的相关性,在程序执行时,用硬件方 法判定
- 可以使得程序代码在其他机器上有效地执行
- · 记分牌的主要思想:允许stall后的指令继续
  - 乱序执行(out-of-order execution) => 乱序完成(out-of-order completion)





## CDC 6600 Scoreboard

## CDC 6600 scoreboard的主要缺陷:

- ・没有定向数据通路
- 指令窗口较小, 仅局限于基本块内的调度
  - 基本块按序执行
- 功能部件数较少
- ・结构冲突时不能发射
- ·WAR相关是通过等待解决的
- · WAW相关时,不会进入Issue阶段



## Acknowledgements

- These slides contain material developed and copyright by:
  - John Kubiatowicz (UCB)
  - Krste Asanovic (UCB)
  - John Hennessy (Standford) and David Patterson (UCB)
  - Chenxi Zhang (Tongji)
  - Muhamed Mudawar (KFUPM)
- UCB material derived from course CS152, CS252, CS61C
- KFUPM material derived from course COE501、COE502