

## 第十七讲

北京航空航天大学

1

## 流水线处理器工程化综合方法

- 流水线工程化综合方法基本流程
- 基础指令集与流水线部件
- 无转发数据通路构造方法
- 转发数据通路构造方法
- 功能部件控制信号构造方法
- 冒险控制信号构造方法

北京航空航天大学计算机学院

## 流水线数据通路建模流程

- S1: 构造每条指令的数据通路
  - 该步骤不需要考虑转发!
- S2: 综合完整数据通路
- S3: 增加转发
- S4: 生成功能部件控制信号
- S5: 生成冒险控制信号

北京航空航天大学计算机学院

## 提纲

- 流水线工程化综合方法基本流程
- 基础指令集与流水线部件
- 无转发数据通路构造方法
- 转发数据通路构造方法
- 功能部件控制信号构造方法
- 冒险控制信号构造方法

北京航空航天大学计算机学院

## 基础指令集与标准流水线

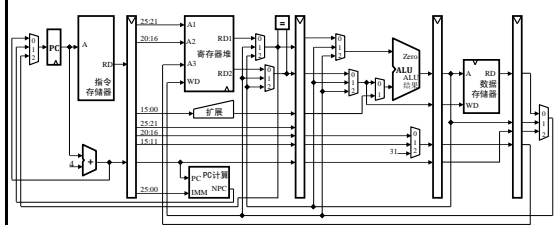
- 指令集
  - lw, sw, addu, subu, ori, lui, beq, j: 典型指令; 可以支持大多数程序需求
  - jal, jalr: 涉及2个写入操作, PC写入, RF写入
    - ◆ 比较特殊的指令

 LW  
SW  
ADDU  
SUBU  
ORI  
LUI  
BEQ  
J  
JAL  
JALR

北京航空航天大学计算机学院

## 基础指令集与标准流水线

- 流水线: 以性能为目标的标准流水线
  - 数据冒险: 转发、暂停
  - 控制冒险: 分支比较前移、转发、暂停

 LW  
SW  
ADDU  
SUBU  
ORI  
LUI  
BEQ  
J  
JAL  
JALR


## 流水线功能部件

- 延用单周期数据通路功能部件
- 按流水段分类，便于理解和记忆
- RF在2个阶段均被使用
  - 译码/读操作数阶段；结果回写寄存器阶段

阶段	部件	输入	输出	控制	描述
取指令	PC	D	Q	PCWr	程序计数器
	ADD4	PC, +4	PC4		完成PC+4
	IM	A	D		指令存储器
译码/读操作数	RF	A1, A2, A3, WD	RD1, RD2	RegWr	寄存器堆
	EXT	I16	IMM32	Extop	立即数扩展
	NPC	PC, I26	NextPC	NPCop	为B类/J计算下条地址
	CMP	D1, D2	Result	CMPop	比较2个数
计算	ALU	A, B	ALU	ALUop	算数/逻辑运算
访存	DM	A, WD	RD	MemWr	数据存储器

## 流水线寄存器

- 需要设置4级流水线寄存器
  - 5级流水线的最后一级寄存器为RF
- 标记X：代表对应流水级需要设置相应寄存器
  - IR：4个流水级均需要
  - ALUOut：仅M级和W级需要

名称	功能	D级	E级	M级	W级
IR	传递指令	X	X	X	X
PC4	下一条指令地址	X	X	X	X
RD1	RF的第1个操作数		X		
RD2	RF的第2个操作数		X	X	
EXT	扩展后的32位立即数		X		
ALUOut	ALU计算结果			X	X
DR	数据存储器结果				X

## 提纲

- 流水线工程化综合方法基本流程
- 基础指令集与流水线部件
- 无转发数据通路构造方法
- 转发数据通路构造方法
- 功能部件控制信号构造方法
- 冒险控制信号构造方法

北京航空航天大学计算机学院

## 流水线数据通路构造表格

- 每级由寄存器和功能部件组成
  - 按流水线5个阶段划分
- X@Y：代表Y阶段的X寄存器
  - IR@W：W级的IR
- PC：出现在3个阶段
  - F级：取指令
  - D级：保存PC+4
  - E级：保存B/J/JAL/JALR的值
- RF：出现在2个阶段
  - D阶段：准备操作数
  - W阶段：回写结果

部件	输入
PC	
ADD4	
IM	
PC	
IR@D	
PC4@D	
RF	A1, A2
EXT	
D1	
D2	
PC4	
I26	
NPC	
PC	
IR@E	
PC4@E	
RD1@E	
RD2@E	
EXT@E	
ALU	A, B
IR@M	
PC4@M	
AO@M	
RD2@M	
DM	A, WD
IR@W	
PC4@W	
AO@W	
DR@W	
RF	A3, WD

## 流水线数据通路构造表格

- 每级由寄存器和功能部件组成
  - 按流水线5个阶段划分
- X@Y：代表Y阶段的X寄存器
  - IR@W：W级的IR
- PC：出现在3个阶段
  - F级：取指令

## 1. PC

- RF：出现在2个阶段
  - D阶段：准备操作数
  - W阶段：回写结果

部件	输入
PC	
ADD4	
IM	
PC	
IR@D	
PC4@D	
RF	A1, A2
EXT	
D1	
D2	
PC4	
I26	
NPC	
PC	
IR@E	
PC4@E	
RD1@E	
RD2@E	
EXT@E	
ALU	A, B
IR@M	
PC4@M	
AO@M	
RD2@M	
DM	A, WD
IR@W	
PC4@W	
AO@W	
DR@W	
RF	A3, WD

## S1: LW的数据通路

- 根据RTL描述建立各级流水线寄存器、功能部件见连接关系
  - LW：5级
- IR必填：因为采用分布式译码
- 指令不涉及的不需要填：如PC4
- X[y]：代表X部件的y域
- IR@D[i16]：D级IR的16位立即数

部件	输入	LW
PC		PC
ADD4		PC
IM		PC
PC		ADD4
IR@D		IM
PC4@D		
RF	A1, A2	IR@D[i16]
EXT		IR@D[i16]
NPC		
PC		
IR@E		IR@D
PC4@E		RF RD1
RD1@E		RF RD1
RD2@E		EXT
EXT@E		RD1@E
ALU	A, B	EXT@E
IR@M		IR@E
PC4@M		IR@E
AO@M		ALU
RD2@M		
DM	A, WD	AO@M
IR@W		IR@M
PC4@W		
AO@W		DM
DR@W		
RF	A3, WD	IR@W[i16]

**S1: 全部指令的数据通路**

部件	输入	LW	SW	ADDI	SUBI	ORI	BEQ	J	JAL	JALR
PC	ADD4	PC	PC	PC	PC	PC	PC	PC	PC	PC
IM	IM	PC	PC	PC	PC	PC	PC	PC	PC	PC
PC	ADD4	PC	PC	PC	PC	PC	PC	PC	PC	PC
IR@D	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM
PC4@D	IM	IM	IM	IM	IM	IM	IM	IM	IM	IM
RF	A1	IR@D[rs]	IR@D[rs]	IR@D[rs]	IR@D[rs]	IR@D[rs]	IR@D[rs]	IR@D[rs]	IR@D[rs]	IR@D[rs]
A2	A2	IR@D[rt]	IR@D[rt]	IR@D[rt]	IR@D[rt]	IR@D[rt]	IR@D[rt]	IR@D[rt]	IR@D[rt]	IR@D[rt]
EXT	EXT	IR@D[16]	IR@D[16]	IR@D[16]	IR@D[16]	IR@D[16]	IR@D[16]	IR@D[16]	IR@D[16]	IR@D[16]
D1	D1									
D2	D2									
CMP	CMP									
NPC	NPC									
PC4	PC4									
I26	I26									
IR@E	IR@E	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D
PC4@E	PC4@E	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D	IR@D
RD1@E	RD1@E	RFRD1	RFRD1	RFRD1	RFRD1	RFRD1	RFRD1	RFRD1	RFRD1	RFRD1
RD2@E	RD2@E	RFRD2	RFRD2	RFRD2	RFRD2	RFRD2	RFRD2	RFRD2	RFRD2	RFRD2
EXT@E	EXT@E	EXT	EXT	EXT	EXT	EXT	EXT	EXT	EXT	EXT
ALU	A	RD1@E	RD1@E	RD1@E	RD1@E	RD1@E	RD1@E	RD1@E	RD1@E	RD1@E
B	B	EXT@E	EXT@E	EXT@E	EXT@E	EXT@E	EXT@E	EXT@E	EXT@E	EXT@E
IR@M	IR@M	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E
PC4@M	PC4@M	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E
AO@M	AO@M	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E
RD2@M	RD2@M	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E	IR@E
DM	DM	AO@M	AO@M	AO@M	AO@M	AO@M	AO@M	AO@M	AO@M	AO@M
WD	WD	RD2@M	RD2@M	RD2@M	RD2@M	RD2@M	RD2@M	RD2@M	RD2@M	RD2@M
IR@W	IR@W	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M
PC4@W	PC4@W	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M
AO@W	AO@W	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M
DR@W	DR@W	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M
RF	A3	IR@W[rt]	IR@W[rt]	IR@W[rt]	IR@W[rt]	IR@W[rt]	IR@W[rt]	IR@W[rt]	IR@W[rt]	IR@W[rt]
WD	WD	DR@W	DR@W	DR@W	DR@W	DR@W	DR@W	DR@W	DR@W	DR@W

## S2: 综合全部指令的数据通路

- 水平方向归并
  - 去除冗余输入来源
- 在每个输入来源个数大于1的输入端前增加1个MUX
  - 注意: 同时需要产生相应的控制信号
- 特例: NPC的i16和i26归并为i26

部件	输入	输入来源	MUX	控制
PC	ADD4	PC		
IM	IM	PC		
PC	ADD4	NPC		
IR@D	IM	RFRD1	M1	PC5d
PC4@D	IM	RFRD1		
RF	A1	IR@D[rs]		
A2	A2	IR@D[rt]		
EXT	EXT	IR@D[16]		
D1	D1	RFRD1		
CMP	CMP	RFRD2		
NPC	PC4	PC4@D		
I26	I26	IR@D[26]		
IR@E	IR@E	IR@D		
PC4@E	PC4@E	IR@D		
RD1@E	RD1@E	RFRD1		
RD2@E	RD2@E	RFRD2		
EXT@E	EXT@E	EXT		
ALU	A	RD1@E		
B	B	EXT@E		
IR@M	IR@M	IR@E		
PC4@M	PC4@M	IR@E		
AO@M	AO@M	IR@E		
RD2@M	RD2@M	IR@E		
DM	DM	AO@M		
WD	WD	RD2@M		
IR@W	IR@W	IR@M		
PC4@W	PC4@W	IR@M		
AO@W	AO@W	IR@M		
DR@W	DR@W	IR@M		
RF	A3	IR@W[rt]		
WD	WD	DR@W		

## 提纲

- 流水线工程化综合方法基本流程
- 基础指令集与流水线部件
- 无转发数据通路构造方法
- 转发数据通路构造方法
- 功能部件控制信号构造方法
- 冒险控制信号构造方法

北京航空航天大学计算机学院

## S3: 增加转发MUX

- 在每个包含了来自RF的输出信息的功能部件的输入点前部署1个MUX(N+1选1)
  - 流水线寄存器除外
  - N: 该级之后的那些能产生计算结果的流水寄存器或MUX
  - 1: RF输出自身
- 3选1 MUX: M级之前的功能部件
  - 输入1: 为RF的输出
  - 输入2: M级计算结果, 即AO@M
  - 输入3: W级计算结果, 即M4
- 2选1 MUX: M级的功能部件(只有DM)

北京航空航天大学计算机学院

## S3.1: 遍历并定位

- 对象: 每个功能部件的每个输入点
  - 包括MUX
  - 流水线寄存器除外
    - 前提: RF支持内部转发!
- 条件: 直接引用或传递引用RF的输出信息

部件	输入	输入来源	MUX
PC	ADD4	PC	
IM	IM	PC	
PC	ADD4	NPC	
IR@D	IM	RFRD1	M1
PC4@D	IM	RFRD1	
RF	A1	IR@D[rs]	
A2	A2	IR@D[rt]	
EXT	EXT	IR@D[16]	
D1	D1	RFRD1	
D2	D2	RFRD2	
CMP	CMP	RFRD2	
NPC	PC4	PC4@D	
I26	I26	IR@D[26]	
IR@E	IR@E	IR@D	
PC4@E	PC4@E	IR@D	
RD1@E	RD1@E	RFRD1	
RD2@E	RD2@E	RFRD2	
EXT@E	EXT@E	EXT	
ALU	A	RD1@E	
B	B	EXT@E	
IR@M	IR@M	IR@E	
PC4@M	PC4@M	IR@E	
AO@M	AO@M	IR@E	
RD2@M	RD2@M	IR@E	
DM	DM	AO@M	
WD	WD	RD2@M	
IR@W	IR@W	IR@M	
PC4@W	PC4@W	IR@M	
AO@W	AO@W	IR@M	
DR@W	DR@W	IR@M	
RF	A3	IR@W[rt]	
WD	WD	DR@W	

## S3.2: 前置MUX

- 在每个输入点前放置1个MUX
- M级: 2选1 MUX
  - 输入0: 原始输入
  - 输入1: W级最终结果
- E级以前(含E): 4选1 MUX
  - 输入0: 原始输入
  - 输入1: M级ALU结果
  - 输入2: M级J类结果
  - 输入3: W级最终结果

部件	输入	输入来源	MUX
PC	ADD4	PC	
IM	IM	PC	
PC	ADD4	NPC	
IR@D	IM	RFRD1	M1
PC4@D	IM	RFRD1	
RF	A1	IR@D[rs]	
A2	A2	IR@D[rt]	
EXT	EXT	IR@D[16]	
D1	D1	RFRD1	
D2	D2	RFRD2	
CMP	CMP	RFRD2	
NPC	PC4	PC4@D	
I26	I26	IR@D[26]	
IR@E	IR@E	IR@D	
PC4@E	PC4@E	IR@D	
RD1@E	RD1@E	RFRD1	
RD2@E	RD2@E	RFRD2	
EXT@E	EXT@E	EXT	
ALU	A	RD1@E	
B	B	EXT@E	
IR@M	IR@M	IR@E	
PC4@M	PC4@M	IR@E	
AO@M	AO@M	IR@E	
RD2@M	RD2@M	IR@E	
DM	DM	AO@M	
WD	WD	RD2@M	
IR@W	IR@W	IR@M	
PC4@W	PC4@W	IR@M	
AO@W	AO@W	IR@M	
DR@W	DR@W	IR@M	
RF	A3	IR@W[rt]	
WD	WD	DR@W	

部件	输入	输入来源	MUX	转发MUX	控制信号	输入1 RF输出	输入2 M级ALU运算	输入3 M级JAL类	输入4 W级最终结果		
PC											
ADD4		PC									
IM		PC									
PC4@D		ADD4	NPC	RF.RD1	M1	MB1	BypassPC_D	RFRD1	AO@M	PC4@M	M4
IR@D		IM									
PC4@D		ADD4									
RF	A1	IR@D[rs]									
A2		IR@D[rt]									
EXT		IR@D[16]									
CMP	D1	RF.RD1				MB2	BypassA_D	RFRD1	AO@M	PC4@M	M4
D2		RF.RD2				MB3	BypassB_D	RFRD2	AO@M	PC4@M	M4
NPC	PC4	PC4@D									
	126	IR@D[126]									
IR@E		IR@D									
PC4@E		PC4@D									
RD1@E		RF.RD1									
RD2@E		RF.RD2									
EXT@E		EXT									
ALU	A	RD1@E				MB4	BypassA_E	RD1@E	AO@M	PC4@M	M4
B		EXT@E	RD2@E			MB5	BypassB_E	RD2@E	AO@M	PC4@M	M4
IR@M		IR@E									
PC4@M		PC4@E									
AO@M		ALU									
RD2@M		RD2@E									
DM	A	AO@M									
WD		RD2@M				MB6	BypassB_M	RD2@M			M4
IR@W		IR@M									
PC4@W		PC4@M									
AO@W		AO@M									
DR@W		DM									
RF	A3	IR@W[rs]	IR@W[rd]	0x1F	M3						

■ 将需转发支持的输入点都用MUX替换

- 将需转发支持的输入点都用MUX替换

## 提纲

- 流水线工程化综合方法基本流程
- 基础指令集与流水线部件
- 无转发数据通路构造方法
- 转发数据通路构造方法
- 功能部件控制信号构造方法
- 冒险控制信号构造方法

北京航空航天大学计算机学院

## 基本方法

- 构造指令集/控制信号矩阵
  - 每条指令占据1行
  - 流水线寄存器、转发MUX：属于冒险控制信号范畴
- 建立指令/控制矩阵关系
  - 用指令的RTL作为指导
  - 给出该RTL执行过程中各功能部件控制信号的取值
  - 填入矩阵一行
- 重复上述过程，完成矩阵构造
- 用真值表方法给出每个控制信号的逻辑表达式

北京航空航天大学计算机学院

## 基本方法

- 构造指令集/控制信号矩阵
  - 每条指令占据1行
  - 流水线寄存器、转发MUX：属于冒险控制信号范畴
- 1. 与单周期工程化方法基本相同
- 2. 逻辑表达式中的**指令**需要替换为功能部件所在流水级的指令寄存器
- 用真值表方法给出每个控制信号的逻辑表达式

北京航空航天大学计算机学院

## 提纲

- 流水线工程化综合方法基本流程
- 基础指令集与流水线部件
- 无转发数据通路构造方法
- 转发数据通路构造方法
- 功能部件控制信号构造方法
- 冒险控制信号构造方法

北京航空航天大学计算机学院

## 冒险分析及处理策略

- 数据冒险
- 控制冒险

北京航空航天大学计算机学院

### 数据冒险：需求级与供给级

- 本质：需求与供给的关系
  - 需求：读操作，寄存器值最终被引用时的位置
  - 供给：写操作，寄存器最新值当前的位置
- 标准流水线存在3个需求级、2个供给级
- 3个需求级：D、E、M
  - B类、JALR：D级
  - 运算类：E级（立即数运算）
  - Store类：E级、M级
- 2个供给级：M、W
  - 运算类指令：结果在M、W
  - LOAD类指令：结果在W

北京航空航天大学计算机学院

### 标准流水线的需求级

- 3个需求级：D、E、M
  - B类、JALR：D级
  - 运算类：E级（立即数运算）
  - Store类：E级、M级
- 注意：需要从需求的位置角度来分类
  - 简单的R/I/J分类不足以指导设计

指令分类	第1操作数需求级	第2操作数需求级
R型算数运算类	E	E
I型算数运算类	E	
LOAD类	E	
STORE类	E	M
B类	D	D
JALR	D	

北京航空航天大学计算机学院

### 标准流水线的供给级

- 3个部件产生寄存器新值：ALU、DM、PC4
  - ALU：E级计算，结果保存在M级，传递给W级
  - DM：M级读取，结果保存在W级
  - PC4：F级计算，结果保存在D级，传递给E/M/W级
    - 但由于JAL类后面的指令被清除，因此
    - 1、当后继指令到达D时，结果已经到达M
    - 2、当后继指令到达E时，结果已经到达W
- 2个供给级：M、W
  - 运算类指令：结果在M、W
  - LOAD类指令：结果在W
  - JAL/JALR：结果在M、W

LW  
SW  
ADDU  
SUBU  
ORI  
LUI  
BEQ  
J  
JAL  
JALR

北京航空航天大学计算机学院

### 从数据冒险处理策略矩阵到代码实现

- S1：构造的转发MUX工作矩阵
- S2：构造数据冒险处理策略矩阵
- S3：将数据冒险处理策略矩阵的行列条件转换为各种变量
- S4：用变量产生MBx的控制信号

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置			LOAD类当前位置			JAL类当前位置		
					E级	M级	W级	E级	M级	W级	E级	M级	W级
MB1	BypassPC_D	D	JALR	RS	暂停	1	3	暂停	暂停	3	不存在	3	不存在
MB2	BypassA_D	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	3	不存在
MB3	BypassB_D	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	3	不存在
MB4	BypassA_E	E	R型计算	RS	--	1	3	--	暂停	3	--	不存在	--
			I型计算	RS	--	1	3	--	暂停	3	--	不存在	--
			LOAD类	RT	--	1	3	--	暂停	3	--	不存在	--
MB5	BypassB_E	E	R型计算	RT	--	1	3	--	暂停	3	--	不存在	--
MB6	BypassB_M	M	STORE类	RT	--	1	3	--	暂停	3	--	不存在	--

### S1：构造MBx工作矩阵

- 注意调整MUX的输入至正确位置
  - 如MB6的M4应调整为输入1的位置

转发MUX	控制信号	位置	读指令	读寄存器	输入0 RF输出	输入1 M级ALU运算	输入2 M级JAL类	输入3 W级最终结果
MB1	BypassPC_D	D	JALR	RS	RFRD1	AO@M	PC4@M	M4
MB2	BypassA_D	D	B类	RS	RFRD1	AO@M	PC4@M	M4
MB3	BypassB_D	D	B类	RT	RFRD2	AO@M	PC4@M	M4
MB4	BypassA_E	E	R型计算	RS	RD1@E	AO@M	PC4@M	M4
			I型计算	RS	RD1@E	AO@M	PC4@M	M4
			LOAD类	RT	RD1@E	AO@M	PC4@M	M4
MB5	BypassB_E	E	R型计算	RT	RD2@E	AO@M	PC4@M	M4
MB6	BypassB_M	M	STORE类	RT	RD2@M	M4		

北京航空航天大学计算机学院

### S2：构造数据冒险策略矩阵

- 水平方向：产生寄存器新值的前序写指令
- 垂直方向：读取寄存器的当前读指令
- 矩阵单元：读指令与写指令出现数据依赖后的对策
  - 如果可以转发，则控制相应MB进行转发
  - 如果无法避免冒险，则暂停

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置			LOAD类当前位置			JAL类当前位置		
					E级	M级	W级	E级	M级	W级	E级	M级	W级
MB1	BypassPC_D	D	JALR	RS	有输出	有输出	有输出	无输出	有输出	有输出	无输出	有输出	有输出
MB2	BypassA_D	D	B类	RS	有输出	有输出	有输出	无输出	有输出	有输出	无输出	有输出	有输出
MB3	BypassB_D	D	B类	RT	有输出	有输出	有输出	无输出	有输出	有输出	无输出	有输出	有输出
MB4	BypassA_E	E	R型计算	RS	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出
			I型计算	RS	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出
			LOAD类	RT	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出
MB5	BypassB_E	E	R型计算	RT	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出
MB6	BypassB_M	M	STORE类	RT	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出	有输出

3大类能够产生寄存器新值的指令

MB4：由于能进入E级的指令均涉及RS，因此目前的分类今后可以合并（即仅仅判断RS即可）

## S2: 构造数据冒险策略矩阵

## ■ 示例: MB2

- 若前序指令为计算类指令
  - E级: 新值正在计算中, 故必须暂停
  - M级: 已有新值, 则BypassA\_D为1
  - W级: 已有新值, 则BypassA\_D为3
- 其他前序指令可同理推演

转发MUX	控制信号	位置	读指令	读寄存器	输入0	输入1	输入2	输入3
MB2	BypassA_D	D	B类	RS	RFRD1	AO@M	PC4@M	M4

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置	E级	M级	W级	E级	M级	W级	E级	M级	W级
MB2	BypassA_D	D	B类	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3

## 数据冒险处理策略矩阵: 示例MB2

- MB2服务指令: B类指令
- 以

转发MUX	位置	连接部件	目标	控制信号	输入0	输入1	输入2	输入3
MB1	D	PC	RS	BypassPC_D	RFRD1	AO@M	PC4@M	M4
MB2	D	比较器	RS	BypassA_D	RFRD1	AO@M	PC4@M	M4
MB3	D	比较器	RT	BypassB_D	RFRD2	AO@M	PC4@M	M4
MB4	E	ALU	RS	BypassA_E	RD1@E	AO@M	PC4@M	M4
MB5	E	ALU	RT	BypassB_E	RD2@E	AO@M	PC4@M	M4
MB6	M	DM	RT	BypassB_M	RD2@M	M4		

转发MUX	位置	读指令	读寄存器	运算类当前位置	LOAD类当前位置	JAL类当前位置	E级	M级	W级	E级	M级	W级	E级	M级	W级
MB2	D	B类	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB3	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3

## 数据冒险策略矩阵

- 暂停: 后级暂停必须联动前级暂停
  - E级发现问题暂停, 同时必须让D级/E级暂停
    - 同时E级插入Nop(结果是在M级体现)
- D级发现问题暂停, 同时必须让F级暂停
  - 同时D级插入Nop(结果是在E级体现)

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置	E级	M级	W级	E级	M级	W级	E级	M级	W级
MB1	BypassPC_D	D	JALR	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB2	BypassA_D	D	B类	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB3	BypassB_D	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB4	BypassA_E	E	R型计算	RS	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			I型计算	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			LOAD类	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB5	BypassB_E	E	R型计算	RT	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB6	BypassB_M	M	STORE类	RT	--	--	1	--	--	1	--	--	1	--	--	1

## 矩阵构造的指令序列案例1

## ■ 序列1:

- I2~I1: 需求E~输出M, 转发
- I3~I1: 需求E~输出W, 转发
- I4~I1: RF转发!!

顺序	指令
I1	sub \$4, \$2, \$3
I2	add \$4, \$1, \$5
I3	or \$6, \$7, \$1
I4	and \$8, \$1, \$9
I5	sw \$10, 0(\$1)

## ■ 序列2:

- I2~I1: 需求M~输出W, 转发

顺序	指令
I1	lw \$1, \$2, \$3
I2	sw \$1, 0(\$4)

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置	E级	M级	W级	E级	M级	W级	E级	M级	W级
MB1	BypassPC_D	D	JALR	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB2	BypassA_D	D	B类	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB3	BypassB_D	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB4	BypassA_E	E	R型计算	RS	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			I型计算	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			LOAD类	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB5	BypassB_E	E	R型计算	RT	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB6	BypassB_M	M	STORE类	RT	--	--	1	--	--	1	--	--	1	--	--	1

## 矩阵构造的指令序列案例2

## ■ 序列3:

- I2~I1: 需求E~输出M, 暂停

顺序	指令
I1	lw \$1, \$2, \$3
I2	sw \$4, 0(\$1)

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置	E级	M级	W级	E级	M级	W级	E级	M级	W级
MB1	BypassPC_D	D	JALR	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB2	BypassA_D	D	B类	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB3	BypassB_D	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB4	BypassA_E	E	R型计算	RS	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			I型计算	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			LOAD类	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB5	BypassB_E	E	R型计算	RT	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB6	BypassB_M	M	STORE类	RT	--	--	1	--	--	1	--	--	1	--	--	1

## 从数据冒险处理策略矩阵到代码实现

- S1: 将行列条件转换为各种变量
- S2: 用变量产生MBX的控制信号
  - MBX: 转发多路选择器
  - 如MB1, MB2

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置	E级	M级	W级	E级	M级	W级	E级	M级	W级
MB1	BypassPC_D	D	JALR	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB2	BypassA_D	D	B类	RS	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB3	BypassB_D	D	B类	RT	暂停	1	3	暂停	暂停	3	不存在	2	3	不存在	2	3
MB4	BypassA_E	E	R型计算	RS	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			I型计算	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
			LOAD类	--	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB5	BypassB_E	E	R型计算	RT	--	1	3	--	暂停	3	--	不存在	3	--	不存在	3
MB6	BypassB_M	M	STORE类	RT	--	--	1	--	--	1	--	--	1	--	--	1

## 数据冒险检测条件

- S1、构造指令的读寄存器
  - D~M: 均涉及读取寄存器操作数
  - 规范表示, 利于理解、编码和发现错误
    - XX\_M代表M级的信号

```
RS_D = IR_D[26:21]
RT_D = IR_D[20:16]
```

北京航空航天大学计算机学院

## 数据冒险检测条件

- S2、建立与读操作的相关变量

D级各种标志

```
BType_D = IR_D[op]==BEQ
JALR_D = IR_D[op]==JALR
```

E级各种标志

```
RType_E = IR_E[op]==000000
ALU_RType = RType@E & (IR_E[fnct]==ADDU | IR_E[fnct]==SUBU)
ALU_IType = IR_E[op]==ORI | IR_E[op]==LUI
MEM_E = IR_E[op]==LW | IR_E[op]==SW
```

M级各种标志

STORE\_M = IR\_M[op]==SW

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置
M01	BypassPC.D	D	JALR	寄存器	寄存器	寄存器	寄存器
M02	BypassA.D	D	寄存器	寄存器	寄存器	寄存器	寄存器
M03	BypassB.D	D	寄存器	寄存器	寄存器	寄存器	寄存器
M04	BypassA.E	E	寄存器	寄存器	寄存器	寄存器	寄存器
M05	BypassB.E	E	寄存器	寄存器	寄存器	寄存器	寄存器

LW  
SW  
ADDU  
SUBU  
ORI  
LUI  
BEQ  
J  
JAL  
JALR

## 数据冒险检测条件

- S3、构造指令的回写目的寄存器及标志
  - M、W级: 产生结果
  - 确保所有指令都被覆盖
    - 计算类、load类、...

```
RD_M = (IR_M[op] == R_TYPE) ? IR_M[rd] :
        IR_M[rt]
```

```
M2RWx_M = (IR_M[op] == LW) ? TRUE : FALSE
A2RWx_M = (IR_M[op] == ADDU) ? TRUE :
            (IR_M[op] == SUBU) ? TRUE :
            (IR_M[op] == ORI) ? TRUE :
            (IR_M[op] == LUI) ? TRUE :
            FALSE
```

北京航空航天大学计算机学院

## 数据冒险转发控制

- 建立所有转发MUX的控制信号条件/取值矩阵
  - 所有的转发都需要有对应的表达式
  - 所有的暂停都留待后续处理

```
BypassBE =
ALU_RType & (RT_E == RD_M) & A2RWx_M == TURE) ? 1 :
ALU_RType & (RT_E == RD_W) & A2RWx_W == TURE) ? 2 :
ALU_RType & (RT_E == RD_W) & M2RWx_W == TURE) ? 2 :
.....
0
```

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置
M01	BypassPC.D	D	JALR	寄存器	寄存器	寄存器	寄存器
M02	BypassA.D	D	寄存器	寄存器	寄存器	寄存器	寄存器
M03	BypassB.D	D	寄存器	寄存器	寄存器	寄存器	寄存器
M04	BypassA.E	E	寄存器	寄存器	寄存器	寄存器	寄存器
M05	BypassB.E	E	寄存器	寄存器	寄存器	寄存器	寄存器

## 数据冒险暂停控制

- 所有暂停
  - PC写使能
  - 流水线写使能
  - 流水线寄存器清除

```
PC写使能 =
ALU_RType & (RT_E == RD_M) & M2RWx_M == TURE) ? 暂停 :
.....
允许
```

转发MUX	控制信号	位置	读指令	读寄存器	计算类当前位置	LOAD类当前位置	JAL类当前位置
M01	BypassPC.D	D	JALR	寄存器	寄存器	寄存器	寄存器
M02	BypassA.D	D	寄存器	寄存器	寄存器	寄存器	寄存器
M03	BypassB.D	D	寄存器	寄存器	寄存器	寄存器	寄存器
M04	BypassA.E	E	寄存器	寄存器	寄存器	寄存器	寄存器
M05	BypassB.E	E	寄存器	寄存器	寄存器	寄存器	寄存器

北京航空航天大学计算机学院

## 冒险统一分析及处理策略

- 数据冒险
- 控制冒险

### 控制冒险的处理策略

- J: 清除后继指令
- JAL: 清除后继指令; 按照R型对待写寄存器
- JALR: 清除后继指令; 按照R型对待写寄存器
- B: 如果条件成立, 则清除后继指令

北京航空航天大学计算机学院

### 检测条件与控制动作

- 检测条件:
  - B类?
  - J/JAL/JALR?
- 控制动作:
  - D级流水线寄存器清除
  - E级流水线寄存器清除(可选)

北京航空航天大学计算机学院

谢谢!

北京航空航天大学计算机学院