

Architecture des microprocesseurs

RISC-V(RV32I)

control de flot : sauts et branchements


Michel Agoyan : michel.agoyan@st.com
Marc Lacruche : marc.lacruche@st.com
Simon Pontie : simon.pontie@cea.fr
Olivier Potin : olivier.potin@emse.fr
Come Allart : come.allart@emse.fr
Raphaël Comps : rcomps@emse.fr



Architecture µP: RV32I saut et branchements

* RISC-V (RV32I) format des instructions


	31 ... 25	24 ... 20	19 ... 15	14 ... 12	11 ... 7	6 ... 0
R	func7	rs2	rs1	func3	rd	0110011
I	imm[11:0]		rs1	func3	rd	0010011
S	imm[11:5]	rs2	rs1	func3	imm[4:0]	0100011
SB	imm[12] imm[10:5]	rs2	rs1	func3	imm[4:1] imm[1]	1100111
U		imm[31:12]			rd	0110111
CJ	imm[20] imm[10:1] imm[19:12]				rd	1101111

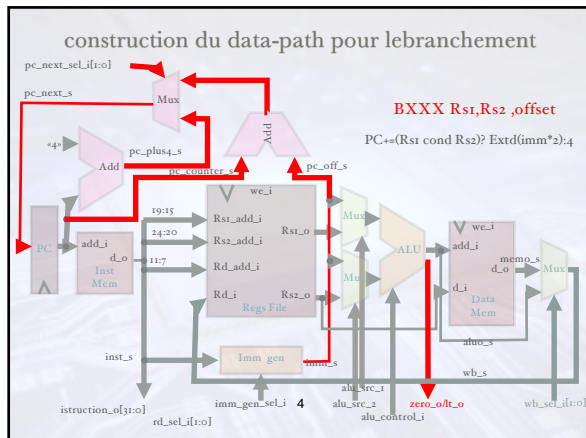


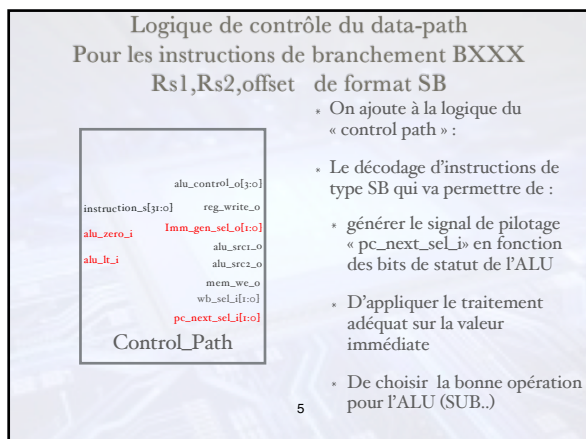
RV32I

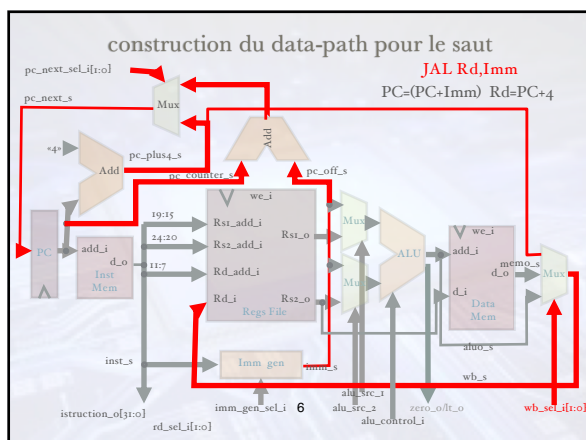
Sauts et Branchements

Architecture monocycle

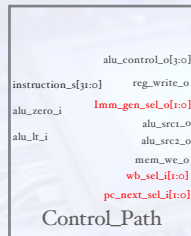








logique de contrôle du data-path Pour les instructions de saut JAL Rd,Imm de de format UJ



- On ajoute à la logique du « control path » :
- Le décodage d'instructions de type UJ qui va permettre de :
 - générer le signal de pilotage « pc_next_sel_i »
- D'appliquer le traitement adéquat sur la valeur immédiate
- De sectionner l'entrée du multiplexeur pour le WB

7

RV32I Sauts et Branchements Architecture pipeline

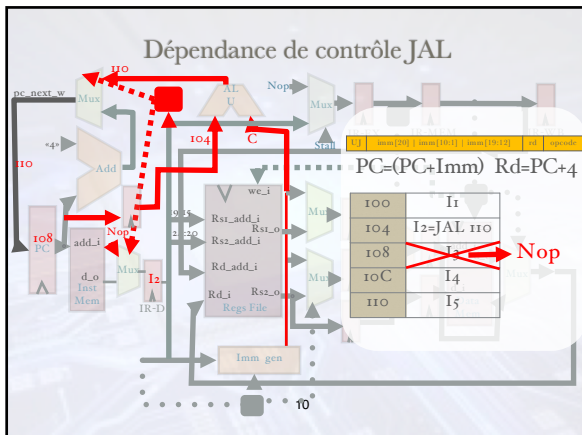


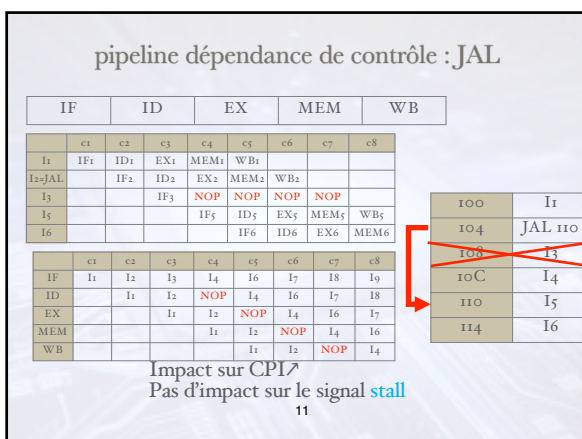
pipeline : dépendance de contrôle

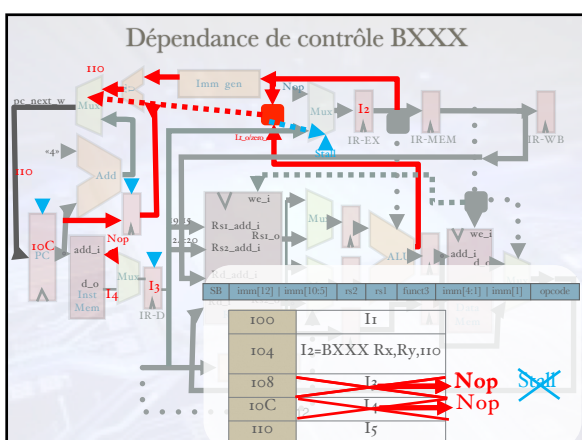
- En cas de rupture du flot d'exécution par une instruction de saut ou de branchement comment est calculée l'adresse de destination (architecture de type pipeline RV32I) ?
- JAL saut à PC + immédiate
- JALR saut indirect à Rs1+ immédiate
- Bxxx si (Rs1 cond Rs2) branchement à PC+Imm

instruction	Étage pipeline pour le calcul de décision	Etage pipeline de calcul d'adresse de destination
JAL	Decode	Decode
JALR	Decode	Execution
Bxxx	Execution	Decode

9







pipeline dépendance de contrôle : BXXX

	IF		ID		EX		MEM		WB	
	c1	c2	c3	c4	c5	c6	c7	c8		
I1	IF1	ID1	EX1	MEM1	WB1					
I2=Bxxx		IF2	ID2	EX2	MEM2	WB2				
I3			IF3	ID3	NOP	NOP	NOP			
I4				IF4	NOP	NOP	NOP	NOP		
I5					IF5	ID5	EX5	MEM5		
	c1	c2	c3	c4	c5	c6	c7	c8		
IF	I1	I2	I3	I4	I5	I6	I7	I8	100	I1
ID		I1	I2	I3	NOP	I5	I6	I7	104	Bxxx Rx,Ry,I10
EX			I1	I2	NOP	NOP	I5	I6	108	I3
MEM				I1	I2	NOP	NOP	I5	112	I4
WB					I1	I2	NOP	NOP	116	I5
									114	I6

Impact sur CPI //

Impact sur le signal Stall:

Par exemple pour BEQ

$$\text{stall} = \text{stall} * ((\text{opcode} == \text{BEQ}) * \text{zero}_o)$$

RV32I

Sauts et Branchements

Architecture pipeline

Optimisation

- Prédiction de branchement (Branch Prediction)
- Table d'adresses de destination (Branch Target Buffer)



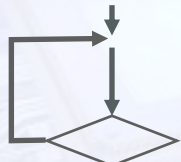
pipeline dépendance de contrôle : optimisation MIPS I

- Les branchements avec test sont utilisés pour les boucles \Rightarrow Impact important sur CPI
- Idée \Rightarrow Ne pas détruire l'instruction qui suit l'instruction de saut ou de branchement = « délai slot »
- Pas forcément une bonne idée du point de vue abstraction : la microarchitecture a un impact sur l'ISA et sur les couches supérieures :
- l'évolution de la microarchitecture (nombre d'étages par exemple du pipeline) devient difficile

LHI R1,0
ORI R1,#1000
...
BNQZ
ADD R1,R1,#-1
...

Prédiction de branchement statique

Probabilité de branchement :- 60% (Benchmark)



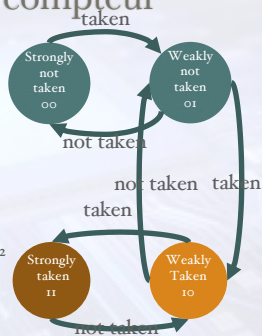
Boucle c:
Branchement Arrière
Probabilité : ~90%



Switch case c:
Branchement Avant
Probabilité : ~50%

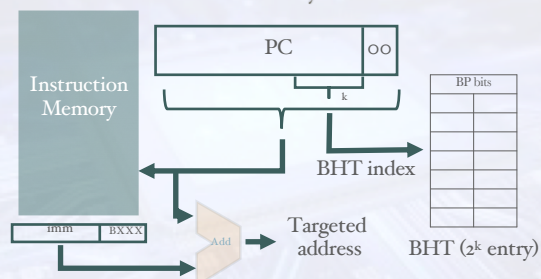
Prédiction de branchement Dynamique: compteur

- Basée sur :
 - Corrélation temporelle
 - Historique d'exécution
 - Corrélation spatiale
 - Chemin d'exécution préférentiel
- On associe 2 BP bits/instruction
- On change le type de prédiction après 2 erreurs consécutives de prédiction



Prédiction dynamique de branchement:

“Branch History Table”



- Une table à 4096 entrées assure ~ 90 % de prédictions justes

Prédiction de branchement dynamique: "Branch Target Buffer"

- On associe à la BTH ("Branch Target History") un BTB ("Branch Target Buffer")
- Dans l'étape de "Fetch" on charge l'instruction issue du mécanisme de prédiction
- Si la prédiction est fausse, on supprime l'instruction chargée
- Dans tous les cas on met à jour BHT et BTB

