

Unité centrale

Briques de base ?

- **UAL** : calculs
- **Registres** : pour calcul, banc avec contrôleur
- **Mémoire** centrale : (instructions + données) banc avec contrôleur

Chemin de données : organisation traitement données et instructions

1. Quel chemin (pour l'archi) ?
2. Quel chef d'orchestre ?

Unité centrale

Qui dit chef d'orchestre...

Synchronisation

pas de conflit

- Circuits (bascules) synchrones au max
- **ICI** transitions registres sur front descendant

comment ?

Stockage : **registre**

- Mémo en fin de cycle + conservation cycle suivant
- Circuit :
 - Entrées branchées sur sorties de registres
 - Sorties branchées sur entrées de registres

Évt le même

Unité centrale

De quoi a-t-on besoin ?

RAM : mémo + adressage, vu à l'épisode précédent

UAL : déjà vu, exemple : sélection par multiplexeur

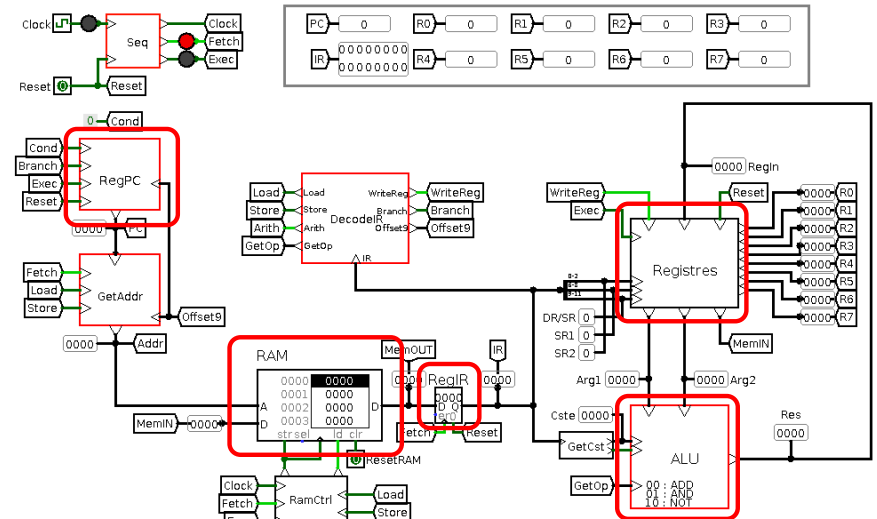
Registres généraux

Programme ~> **Registres spéciaux** IR, PC

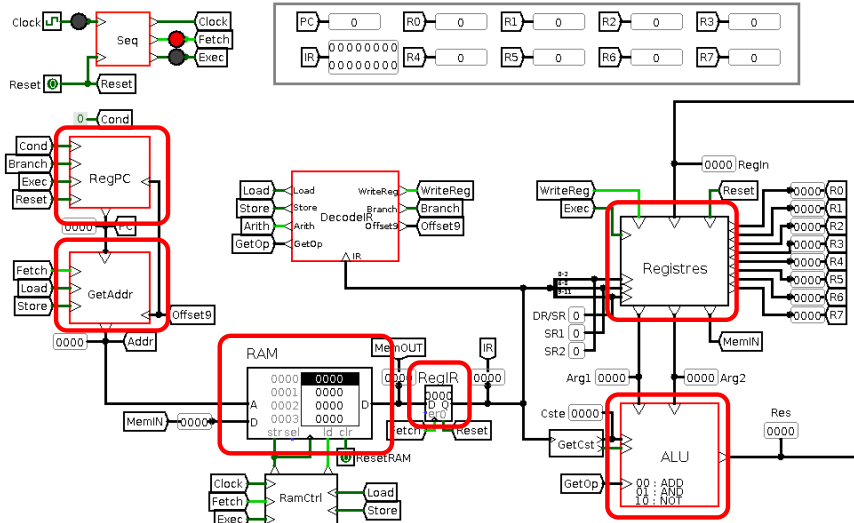
...

Circuits

Unité centrale



Unité centrale



Unité centrale

Cycle **instruction** ≠ cycle **horloge**

Comp : opérands lues dans registres, rangement résultat à la fin

Mem : liaison registre/mémoire, écriture à la fin

BR : calcul adresse et écriture dans PC à la fin

Unité centrale

Cycles

Cycle **instruction** ≠ cycle **horloge**

Comp : opérands lues dans registres, rangement résultat à la fin

Mem : liaison registre/mémoire, écriture à la fin

BR : calcul adresse et écriture dans PC à la fin

MAIS instruction qq. part, ici dans RI

bien câblé

Ici : Mémoire → IR puis exécution

Cycle d'instruction :

1. **LI** (Fetch)
2. **EX** (Exec)

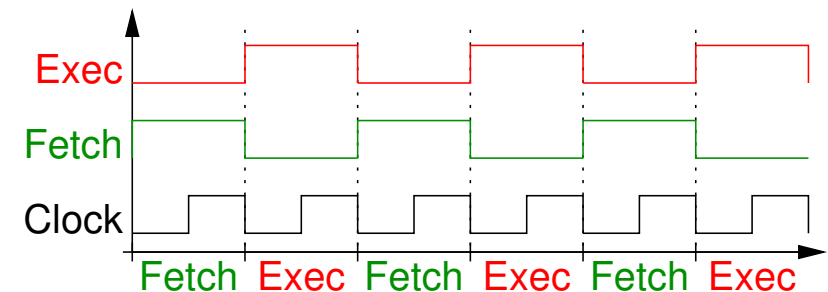
Unité centrale

Synchro

Différentes activités suivant LI ou EX ~ *séquenceur*

in : Clk, (RAZ)

out : Clk, **LI**, **EX**



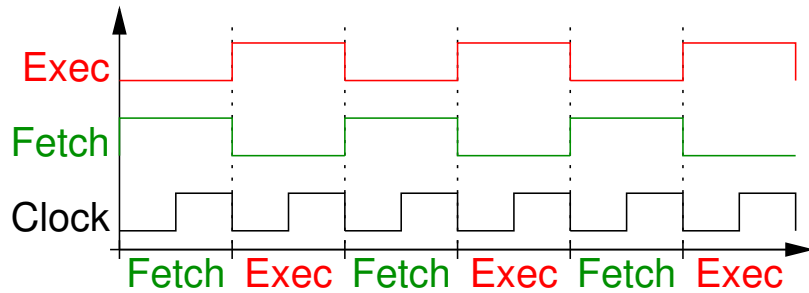
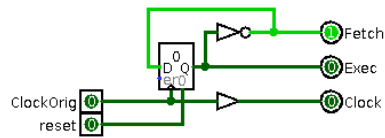
Unité centrale

Synchro

Différentes activités suivant LI ou EX ~ séquenceur

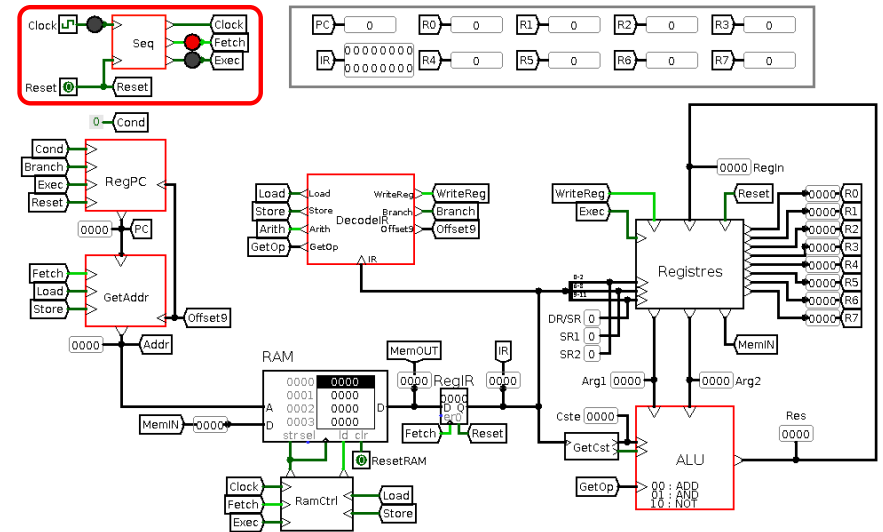
in : Clk, (RAZ)

out : Clk, LI, EX



Unité centrale

Synchro



Unité centrale

Dialogue avec registres

syntaxe	action	NZP	codage															
			opcode				arguments											
			F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
NOT DR,SR	DR <- not SR	*	1	0	0	1	DR	SR			1	1	1	1	1	1		
ADD DR,SR1,SR2	DR <- SR1 + SR2	*	0	0	0	1	DR	SR1	0	0	0					SR2		
ADD DR,SR1,Imm5	DR <- SR1 + SEXT(Imm5)	*	0	0	0	1	DR	SR1	1						Imm5			
AND DR,SR1,SR2	DR <- SR1 and SR2	*	0	1	0	1	DR	SR1	0	0					SR2			
AND DR,SR1,Imm5	DR <- SR1 and SEXT(Imm5)	*	0	1	0	1	DR	SR1	1						Imm5			
LD DR,label	DR <- mem[PC + SEXT(PCoffset9)]	*	0	0	1	0	DR								PCoffset9			
ST SR,label	mem[PC + SEXT(PCoffset9)] <- SR		0	0	1	1	SR								PCoffset9			
BR[n][z][p] label	Si (cond) PC <- PC + SEXT(PCoffset9)		0	0	0	0	n	z	p						PCoffset9			

Qui touche aux registres ?

Comment ?

schéma RISC

Unité centrale

Dialogue avec registres

Adressage pertinent

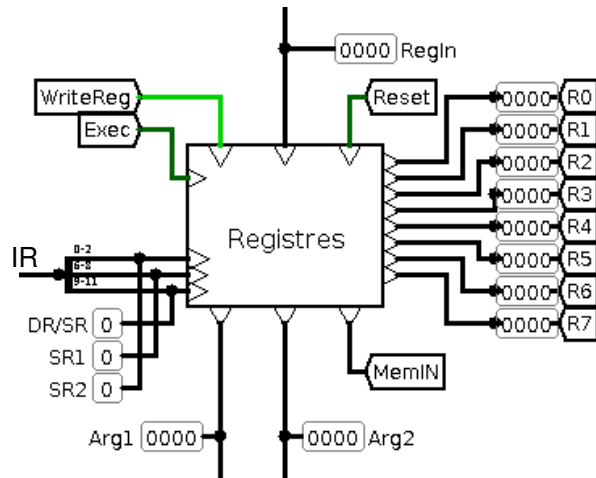
Sorties pertinentes

Entrées pertinentes

Et signaux...

Unité centrale

Dialogue avec registres



Unité centrale

Dialogue avec UAL

syntaxe	action	NZP	codage																	
			opcode				arguments													
			F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0		
NOT DR,SR1	DR ← not SR1	*	1	0	0	1	DR		SR1		1						1			
ADD DR,SR1,SR2	DR ← SR1 + SR2	*	0	0	0	1	DR		SR1		0		0		SR2					
ADD DR,SR1,Imm5	DR ← SR1 + SEXT(Imm5)	*	0	0	0	1	DR		SR1		1		Imm5							
AND DR,SR1,SR2	DR ← SR1 and SR2	*	0	1	0	1	DR		SR1		0		0		SR2					
AND DR,SR1,Imm5	DR ← SR1 and SEXT(Imm5)	*	0	1	0	1	DR		SR1		1		Imm5							

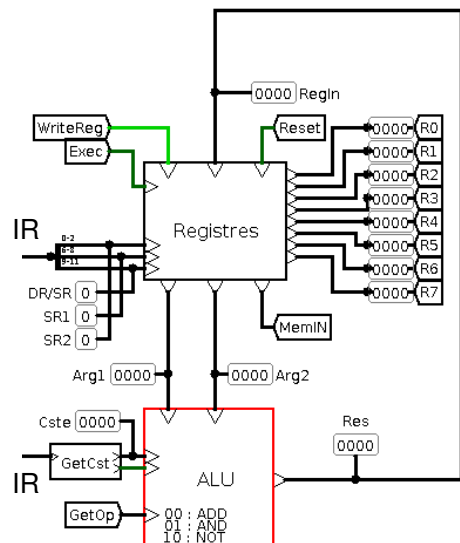
Quels argument ?

- Dest, Source₁, Source₂
- Dest, Source₁, Immédiat
- Dest, Source₁

Entrées/Sorties/Signaux de l'UAL ?

Unité centrale

Dialogue avec UAL



Unité centrale

Dialogue avec mémoire

syntaxe	action	NZP	codage															
			opcode				arguments											
			F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
LD DR,label	DR ← mem[PC + SEXT(PCOffset9)]	*	0	0	1	0	DR		PCOffset9									
ST SR,label	mem[PC + SEXT(PCOffset9)] ← SR		0	0	1	1	SR		PCOffset9									

Avec qui ? Quand ?

- LI et EX
- PC + registres généraux...

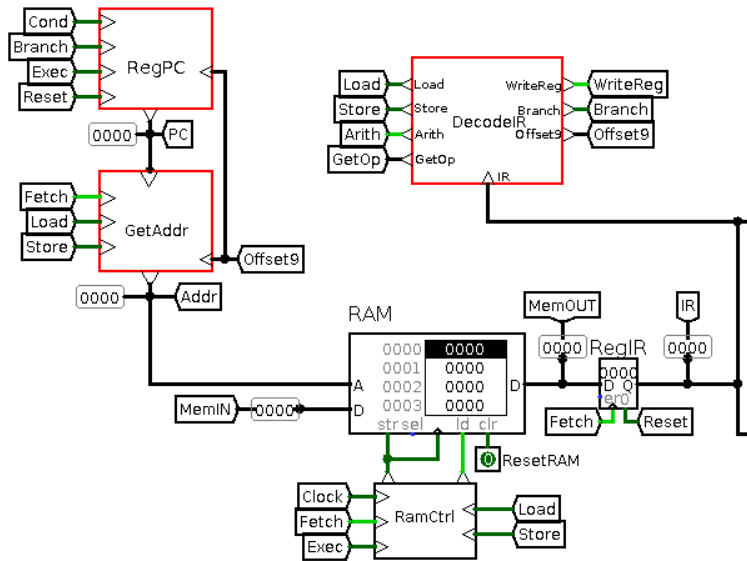
Signaux : Load/Store, Fetch, Exec, Clk

Calculs → GetAddr

PC ou PC+...

Unité centrale

Dialogue avec mémoire



XU - UCBL1 - ASR4 2018/2019

UN JEU DE SLIDES N'EST PAS UN POLY DE RÉFÉRENCE 130

Unité centrale

Dialogue avec IR (décodage)

		opcode				arguments													
		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0		
NOT DR,SR	DR <- not SR	*	1	0	0	1	DR	SR											
ADD DR,SR1,SR2	DR <- SR1 + SR2	*	0	0	0	1	DR	SR1	0	0									SR2
ADD DR,SR1,Imm5	DR <- SR1 + SEXT(Imm5)	*	0	0	0	1	DR	SR1	1										Imm5
AND DR,SR1,SR2	DR <- SR1 and SR2	*	0	1	0	1	DR	SR1	0	0									SR2
AND DR,SR1,Imm5	DR <- SR1 and SEXT(Imm5)	*	0	1	0	1	DR	SR1	1										Imm5
LD DR,label	DR <- mem[PC + SEXT(PCOffset9)]	*	0	0	1	0	DR												PCOffset9
ST SR,label	mem[PC + SEXT(PCOffset9)] <- SR		0	0	1	1	SR												PCOffset9
BR[n][z][p] label	Si (cond) PC <- PC + SEXT(PCOffset9)		0	0	0	0		n	z	p									PCOffset9

Instruction dans IR → Quels signaux pour quelle instruction ?

- GetOp ?
- Load ?
- Store ?
- WriteReg ?

XU - UCBL1 - ASR4 2018/2019

UN JEU DE SLIDES N'EST PAS UN POLY DE RÉFÉRENCE 131

Unité centrale

Organisation

LI

Charger Mem[PC] dans IR

Signal Fetch = 1 → mémoire mode « lecture »

PC directement accessible

IR reçoit MemOUT → écriture et verrou quand Fetch → 0

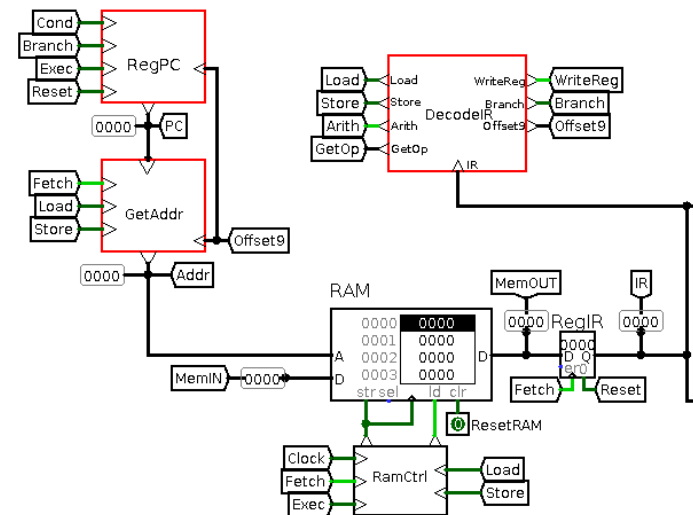
XU - UCBL1 - ASR4 2018/2019

UN JEU DE SLIDES N'EST PAS UN POLY DE RÉFÉRENCE 132

Unité centrale

Organisation

LI



XU - UCBL1 - ASR4 2018/2019

UN JEU DE SLIDES N'EST PAS UN POLY DE RÉFÉRENCE 133

Unité centrale

EX

Mise à jour PC

Opération...

Rangement résultat (à la fin)

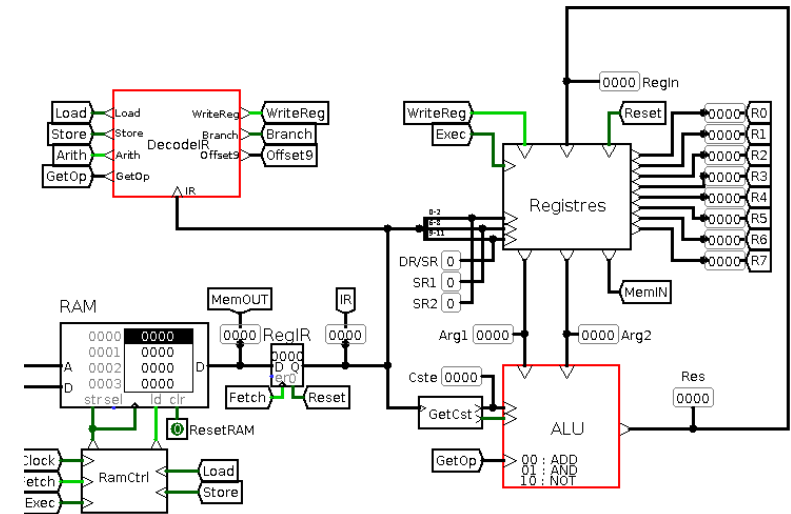
Opération : calcul ou mémoire

Organisation

on sait faire

Unité centrale

EX calcul (exemple ADD)



Unité centrale

EX mémoire (exemple ST)

Organisation

