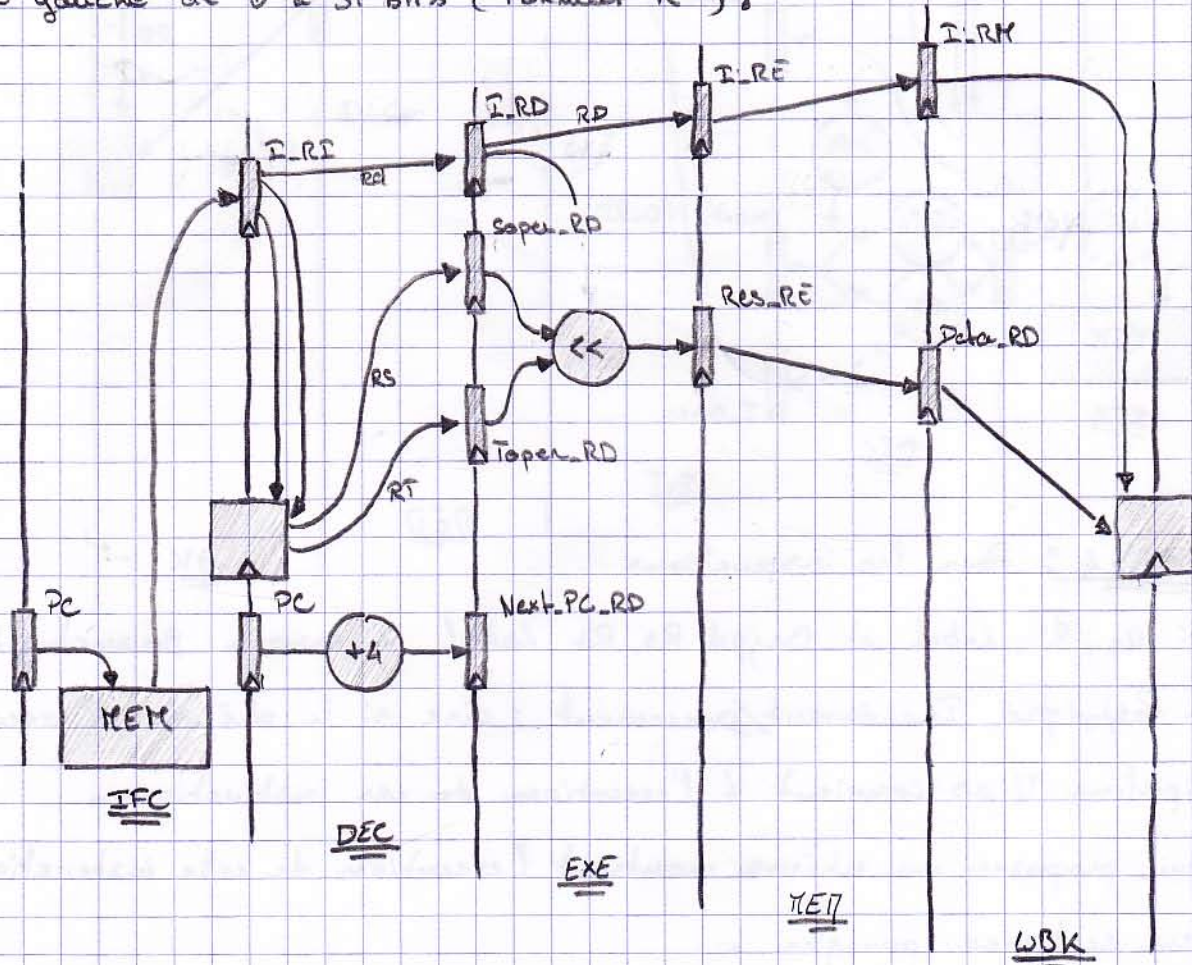


Exe 1 : Montrer à l'aide d'un schéma détaillé l'exécution de l'instr.

SHLV Rd, Rt, Rs dans le pipeline du Mips 32 :

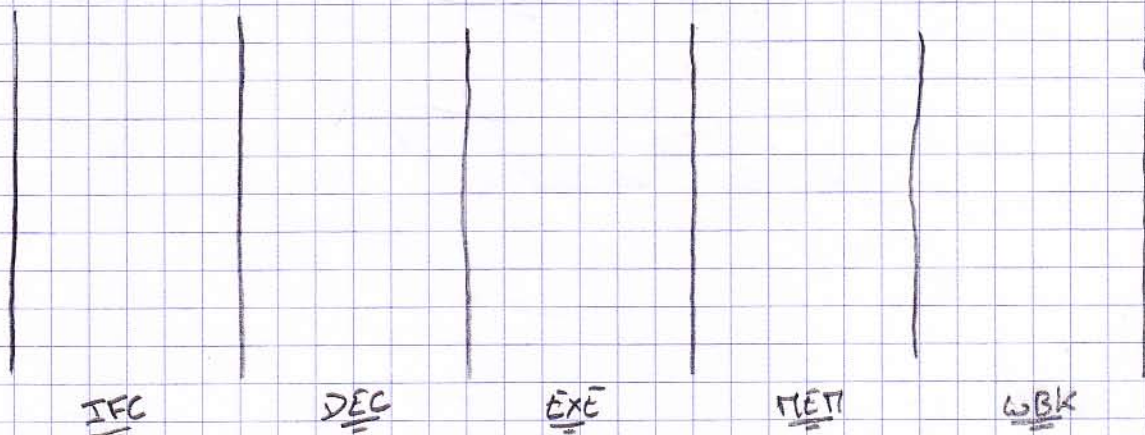
Slvr Rd, Rt, Rs les 5 dernier bits de Rs sont significatif ; Rt est décalé à gauche de 0 à 31 bits (Format R).



Exe 2 : Montrer à l'aide d'un schéma détaillé l'exécution de l'instr.

BLTZAL Rs, label dans le pipeline du Mips 32 :

BLTZAL Rs, label : Branch if Less Than Zero And Link with the fact de stocker l'adresse de retour dans le registre R31.

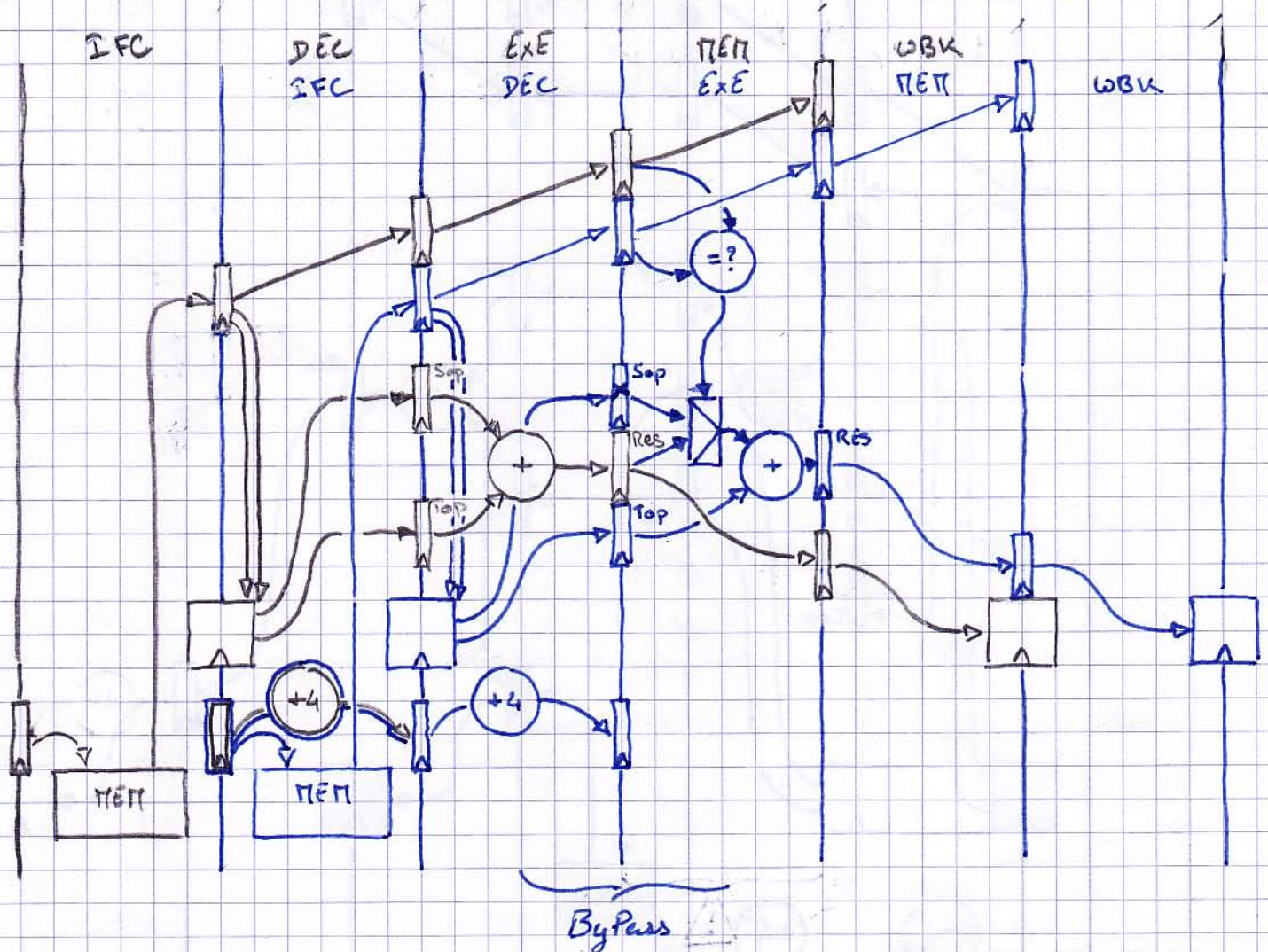
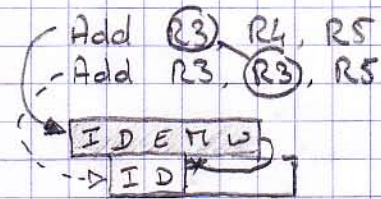


Exe 5 : Montrer à l'aide d'un schéma détaillé l'exécution de la suite d'instruction suivante dans le pipeline de Pip3 32 :

Add R3, R4, R5
Add R3, R3, R5

Quelle est la condition qui déclenche le bypass ?

Problème de dépendance détecté par le matériel qui va déclencher les éléments du Bypass.



Exe 6 : Faut-il faire un Bypass pour la suite

{ Add R0, R3, R1
Add R3, R0, R1 ?

Quelle est la condition déclenchant le Bypass ? Non R0 n'est égal à 0.

Un Bypass se déclenche si le numéro du registre RS(RT) de l'instruction courante (en I-RO) est identique au numéro du registre RD de l'instruction précédente (I-RE) et qu'il ne s'agit pas du numéro 0. Il faut que l'instruction précédente écrive dans RD et l'instr. courante lit RS(RT).

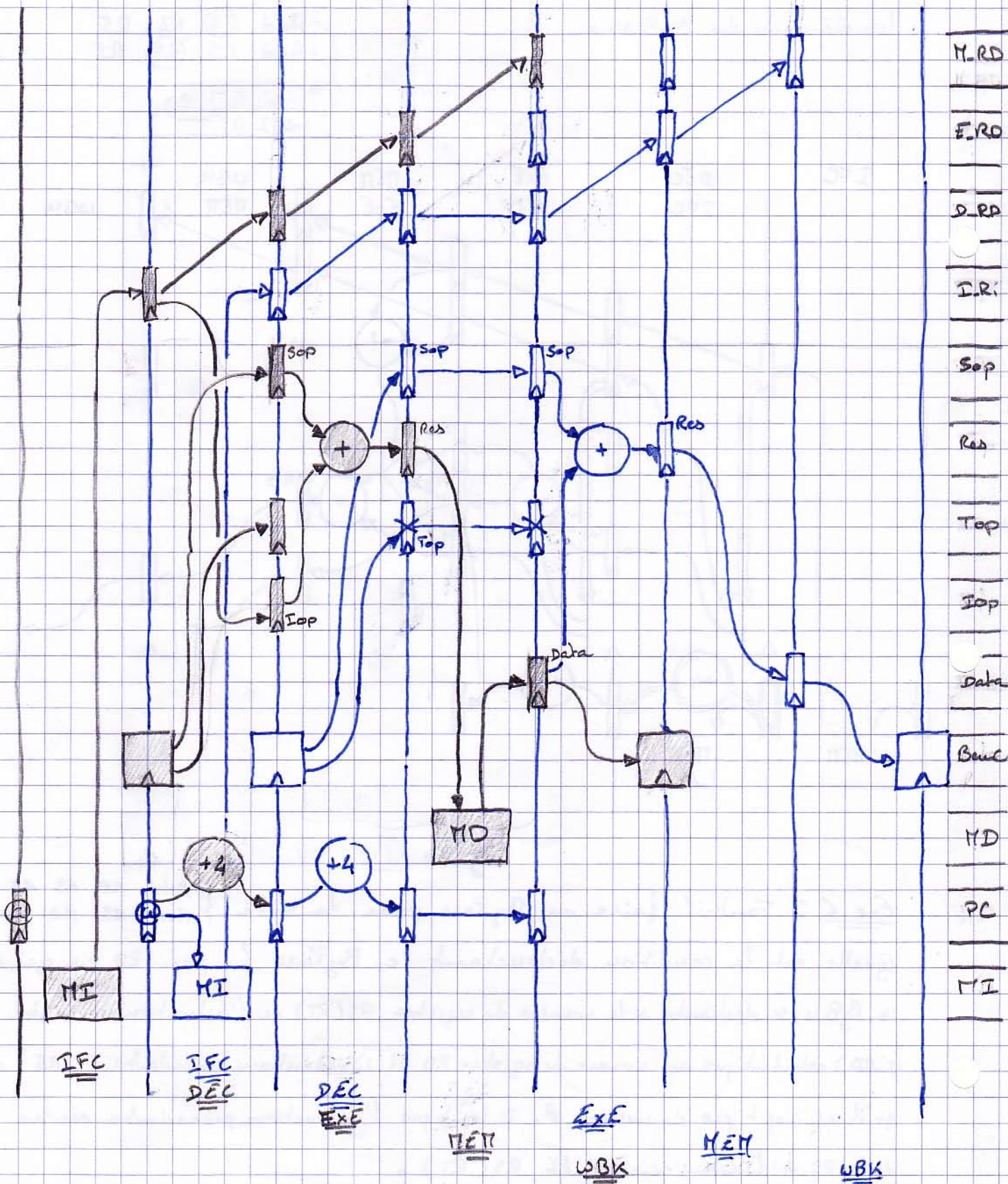
Ex7 2.1002: Montrer à l'aide d'un schéma détaillé l'exécution de la suite d'instruction suivante dans le pipeline du TIPS 32 :

Les $(R3), 0(R2)$ Quelle est la condition déclenchant le By pass ?
 Add $R3, (R3), R1$

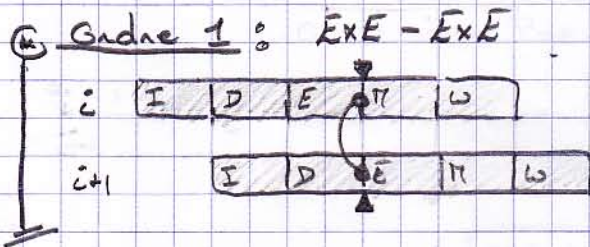
Nop \Rightarrow Sll $R0, R0, 0$

I D E M W

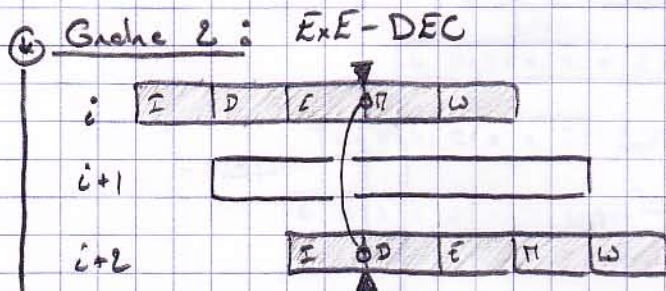
I D E M W



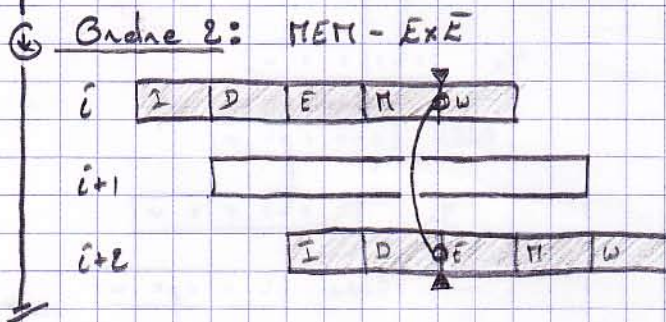
Exe 9 : la réalisation du processeur Mips-32 nécessite la mise en oeuvre de 8 bypass (raccourcis). Pour chacun de ces bypass proposer un exemple de suite d'instructions :



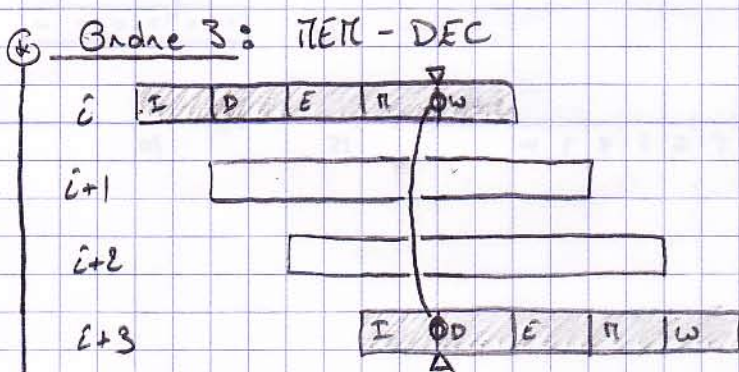
Add (R3), R4, R5 } RD et RS
Add R3, (R3), R5 }
Add (R3), R4, R5 }
Add R3, R5, (R3) } RD et RT



Add (R16), R17, R18 } RD-RS
Nop }
Beg (R16), R0, Label }
Add (R16), R17, R18 } RD-RT
Nop }
Beg R0, (R16), Label }



Lw (R3), 0(R4) } RT-RS
Nop }
Add R4, (R3), R5 }
Lw (R3), 0(R4) } RT-RT
Nop }
Add R4, R5, (R3) }



Lw (R3), 0(R4) } RT-RS
Nop }
Nop }
Beg (R3), R5, Label }
Lw (R3), 0(R4) } RT-RT
Nop }
Nop }
Beg R5, (R3), Label }

Add (R3), R4, R5 } RD-RT
Nop }
Nop }
Add R6, R7, (R3) }

Add (R3), R4, R5 } RD-RS
Nop }
Nop }
Add R6, (R3), R7 }

Exe 8 : Analyser l'exécution de la suite d'instructions suivante dans le processeur

Mips 32 pipeline à l'aide d'un schéma simplifié sans modifier le code "→ Chaos".

Init :

Addiu R2, R0, 4

Lui R3, 0x000C

Add R2, R2, R2

Gui R3, R3, 0x2568

Lw R2, 0(R3)

Lb R2, 0(R2)

Gui R2, R2, 0x0001

Bltz R2, suite

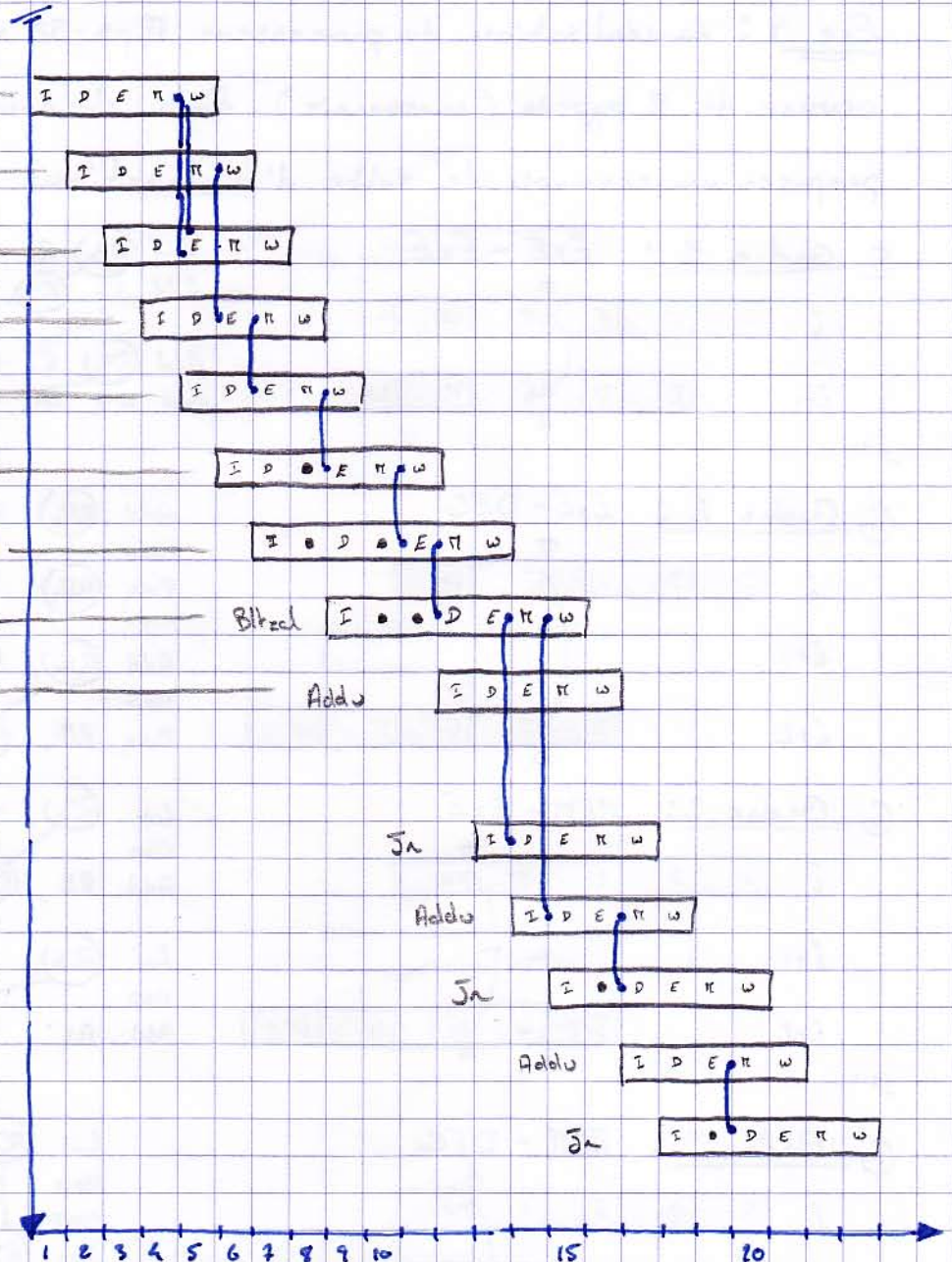
Addu R0, R0, 0

Suite :

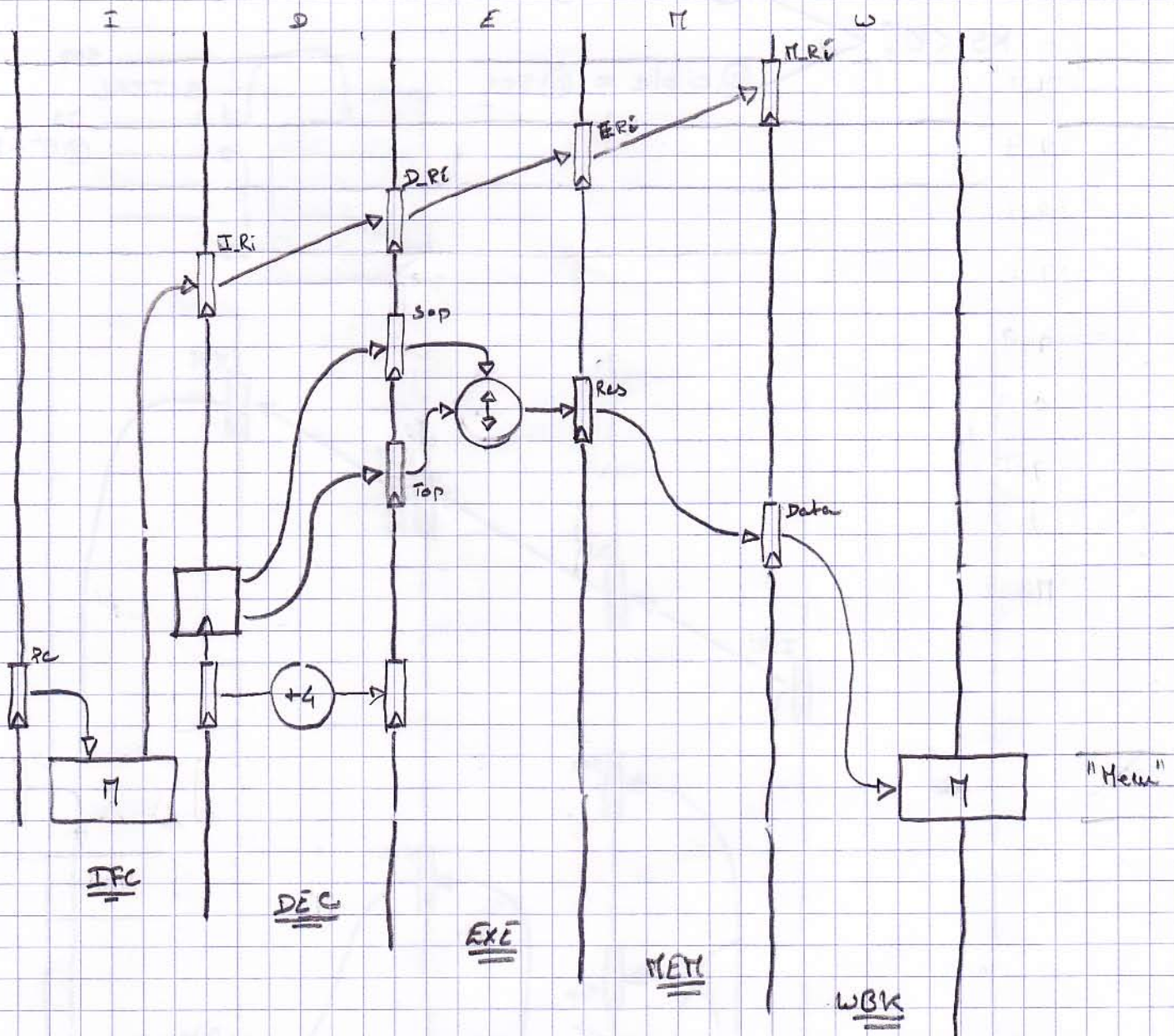
Jn R31

Addu R31, R31, -8

! Infinite loop



Exe 1 by Pinoux : Schéma détaillé de l'exécution de $SLW\ R_d, R_r, R_s$

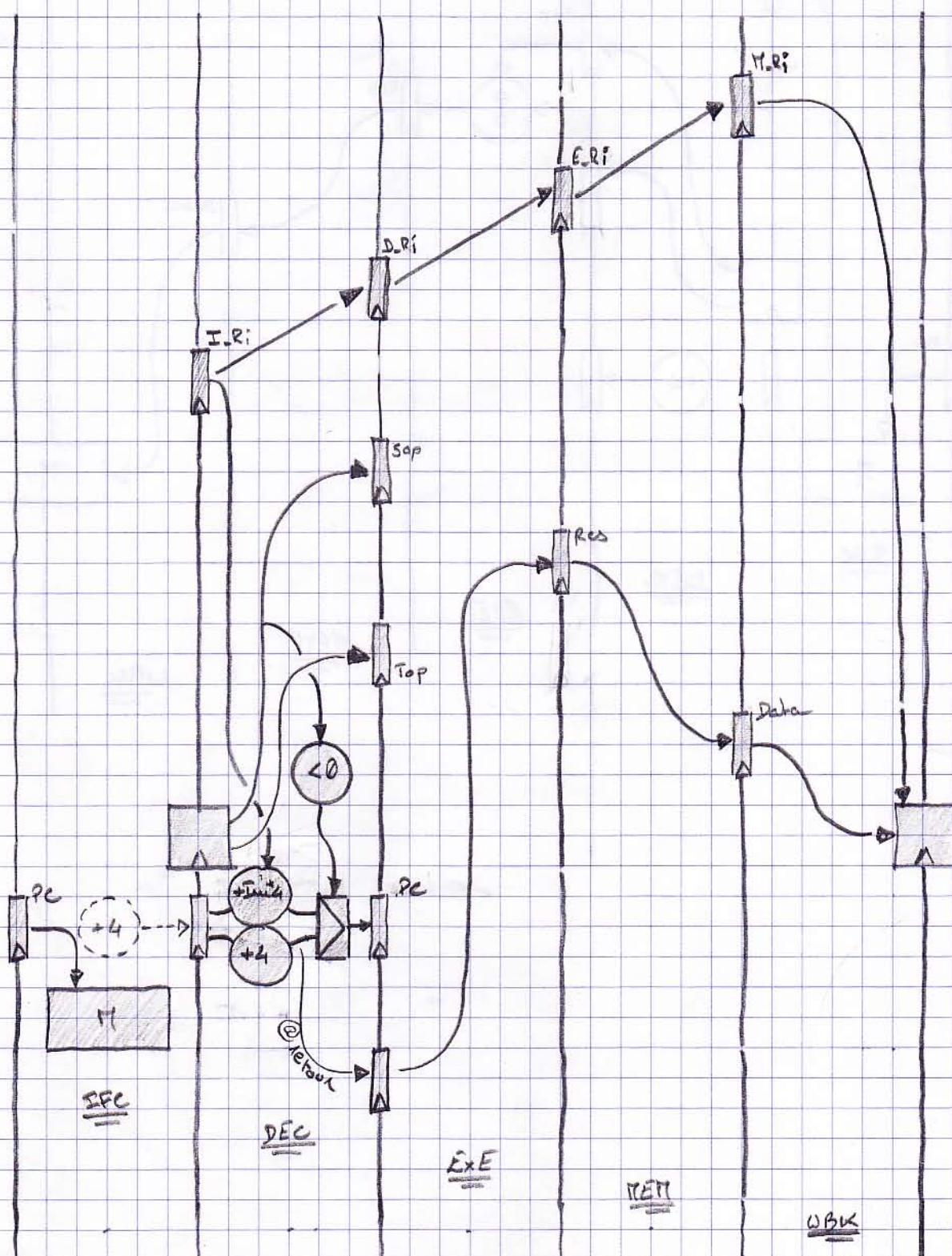
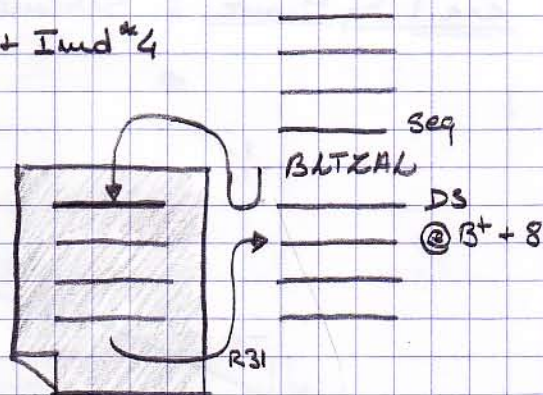


Ex 2 by Pinoux : Schéma détaillé de l'exécution de BLTZAL RS, Label

Format I



$RS < 0?$
 $\begin{cases} @cible = @B^+ + 4 + Imm * 4 \\ @cible = @seq \end{cases}$



Exe 10 by Pinouz : Transformer le code de telle façon qu'il puisse être exécuté sur le Pi ps pipeline à 5 étages.

Analyser l'exécution de la boucle avec un schéma simplifié.

Calculer le nombre de cycles nécessaire à l'exécution d'une itération.

Calculer le CPI et le CPI utile.

Stalen: "version 1"

Add R2, R0, R4

while :

$$1 \quad | \quad L_6 \quad (R_9) \quad \theta(R_4)$$

2 Beg (R9, R0, End

3				Nop
---	--	--	--	-----

4 Addio $R_4, R_4, 1$

5 | Addi R2, R2, 1

6 5 white

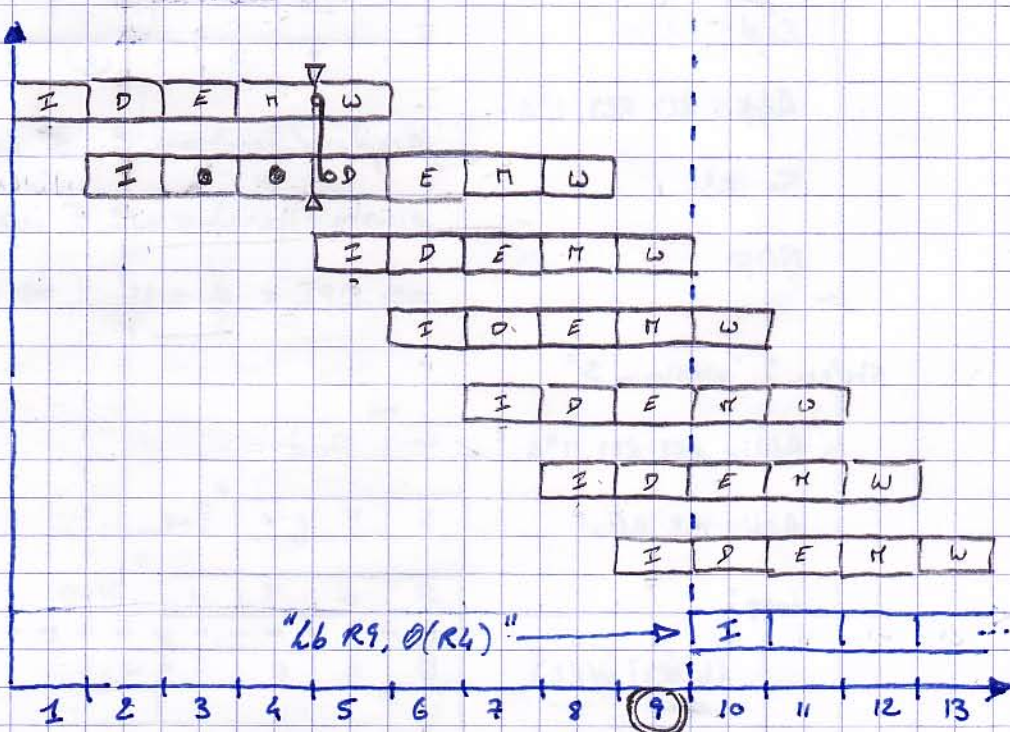
7 NOP

EndWhile:

Jm R31

Ngp

Schéma simplifié de l'exécution de la boucle



#cycles/iteration = 9 cycles

$$\# \text{instruct}^{\circ} / \text{iterat}^{\circ} = 7 \text{ instruct}^{\circ}$$

$$\Rightarrow \text{CPI} = \frac{9}{7} = 1,3 \text{ cycles/instruct}^e$$

$$\Rightarrow \text{CPI}_{\text{utile}} = \frac{9}{5} = 1,8 \text{ cycles/instruction utiles}$$

↳ Utilisation du processeur

statem: "Version 2"

Addio R29, R29, -1*4

Addio R2, R0, R0

loop:

Lb (R9) 0(R4)

Addio R4, R4, 1

Addio R2, R2, 1

Bne (R9), R0, End

Nop

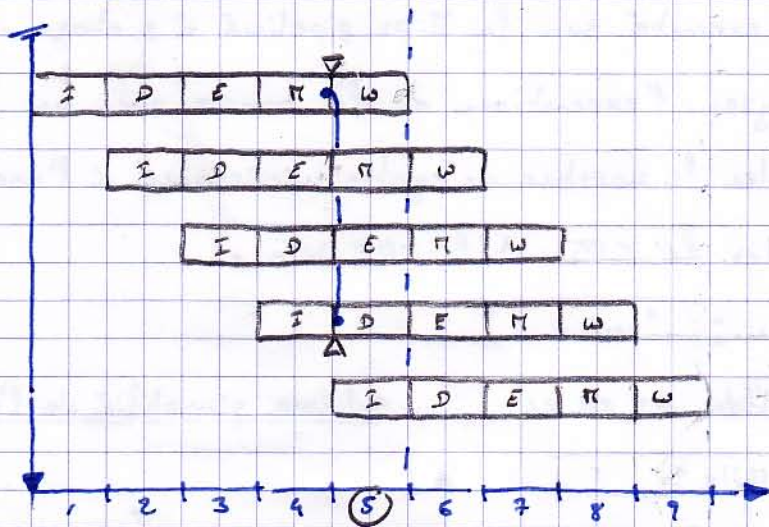
End:

Addio R29, R29, 1*4

Jn R31

NOP

Schema simplifié de l'exécution de la boucle



#cycles/iteration: 5 cycles

#instr/iteration: 5 instructions

$$\Rightarrow CPI = \frac{5}{5} = 1 \quad \Rightarrow CPI_{while} = \frac{5}{4}$$

statem: "version 3"

Addio R29, R29, -1*4

Addio R8, R4, 1

loop:

Lb (R9) 0(R4)

Addio R4, R4, 1

Bne (R9), R0, loop

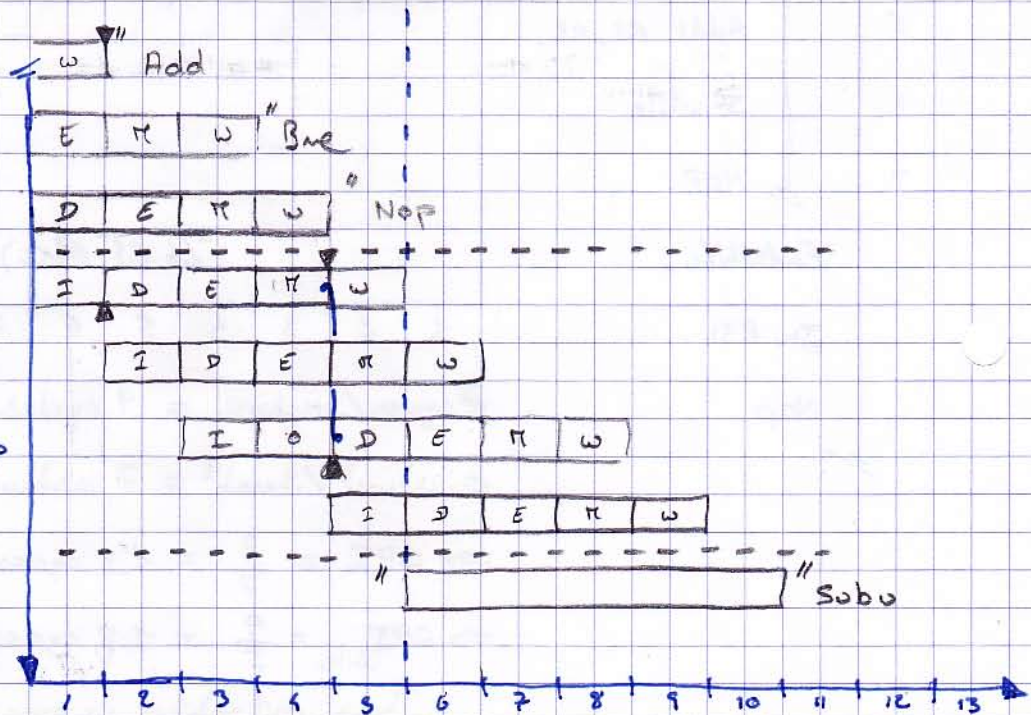
Nop

Subo R2, R4, R8

Addio R29, R29, R29

Jn R31

Nop



#cycles/iteration: 5

#instr/iteration: 4

$$\Rightarrow CPI = \frac{5}{4} \quad \Rightarrow CPI_{while} = \frac{5}{3}$$

Exe II by Pinoux : Transformer le code de telle façon qu'il puisse être exécuté sur le MIPS pipeliné à 5 étages.

Analysen l'exécution de la boucle avec un schéma simplifié.

Calculer le nombre de cycles nécessaire à l'exécution d'une itération.

Calculer le CPI et le CPI_{utile} sachant que 30% des char sont minuscules.

Strupper :

Addiu R29, R29, -4

Addiu R2, R0, R4

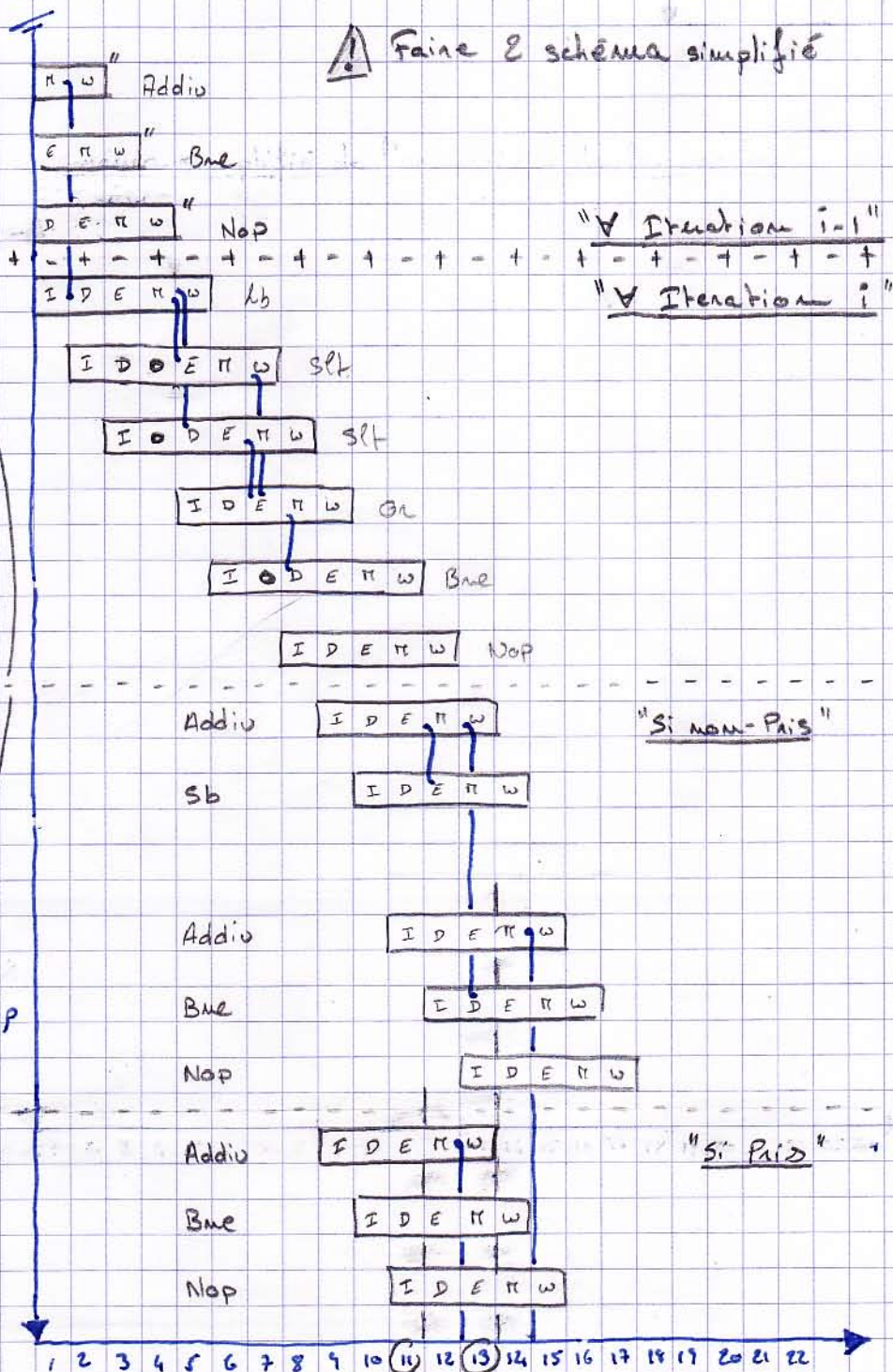
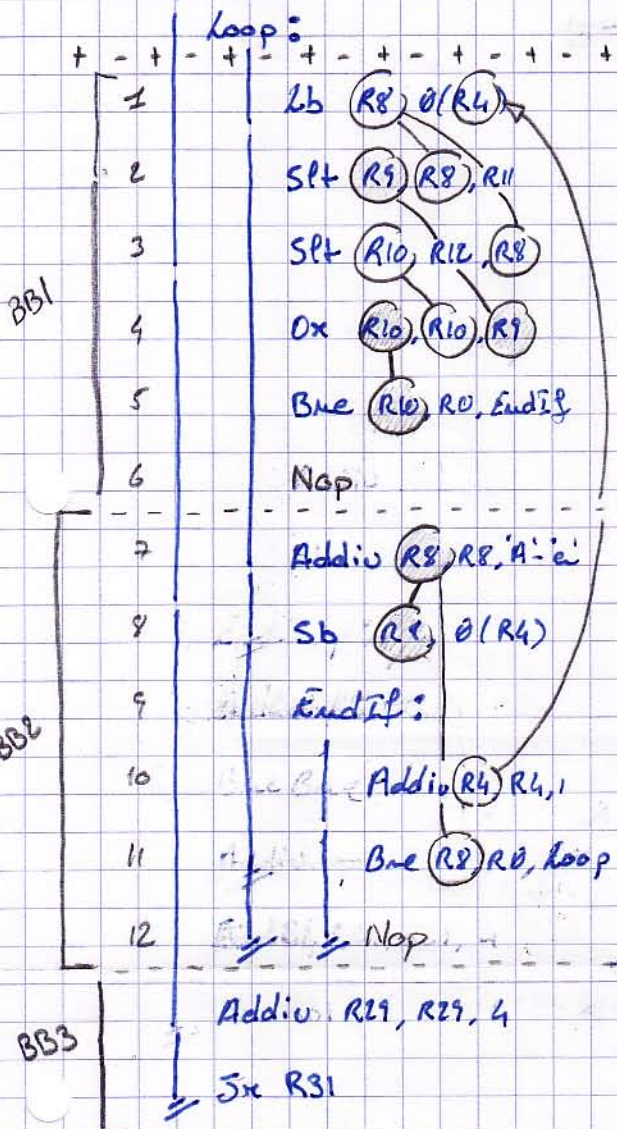
Addiu R11, R0, 'a'

Addiu R12, R0, 'z'

Loop :

Schema simplifié de l'exécution de la boucle

! Faire 2 schéma simplifié



④ Branchement Échoue (30%)

#cycles/iteration = 13 cycles

#instruct°/iteration = 11 instruct°

④ Branchement Réussi (70%)

#cycles/iteration = 11 cycles

#instruct°/iteration = 9 instruct°

④ Le CPI Moyen

$$CPI = \frac{0,3 \times 13 + 0,7 \times 11}{0,3 \times 11 + 0,7 \times 9} \approx 1,208 \text{ cycles/instructions}$$

$$CPI_{\text{utile}} = \frac{0,3 \times 13 + 0,7 \times 11}{0,3 \times 9 + 0,7 \times 7} \approx 1,526 \text{ cycles/instructions utile}$$

Exe 18: Comme à l'exe 11 mais avec la fonction p.g.c.d.

pgcd : Bes R4, R5, End

Nop

Loop: Sltw (R8, R4, R5)

Be, (R8), R0, Else

Nop

Subo RS, RS, R4

J End If

Ngp

Else : Subu R_4, R_4, R_5

End If: Bme R4, R5, Loop

Nop

End : Add R2, R4, R0

5. BSI

Nop

