

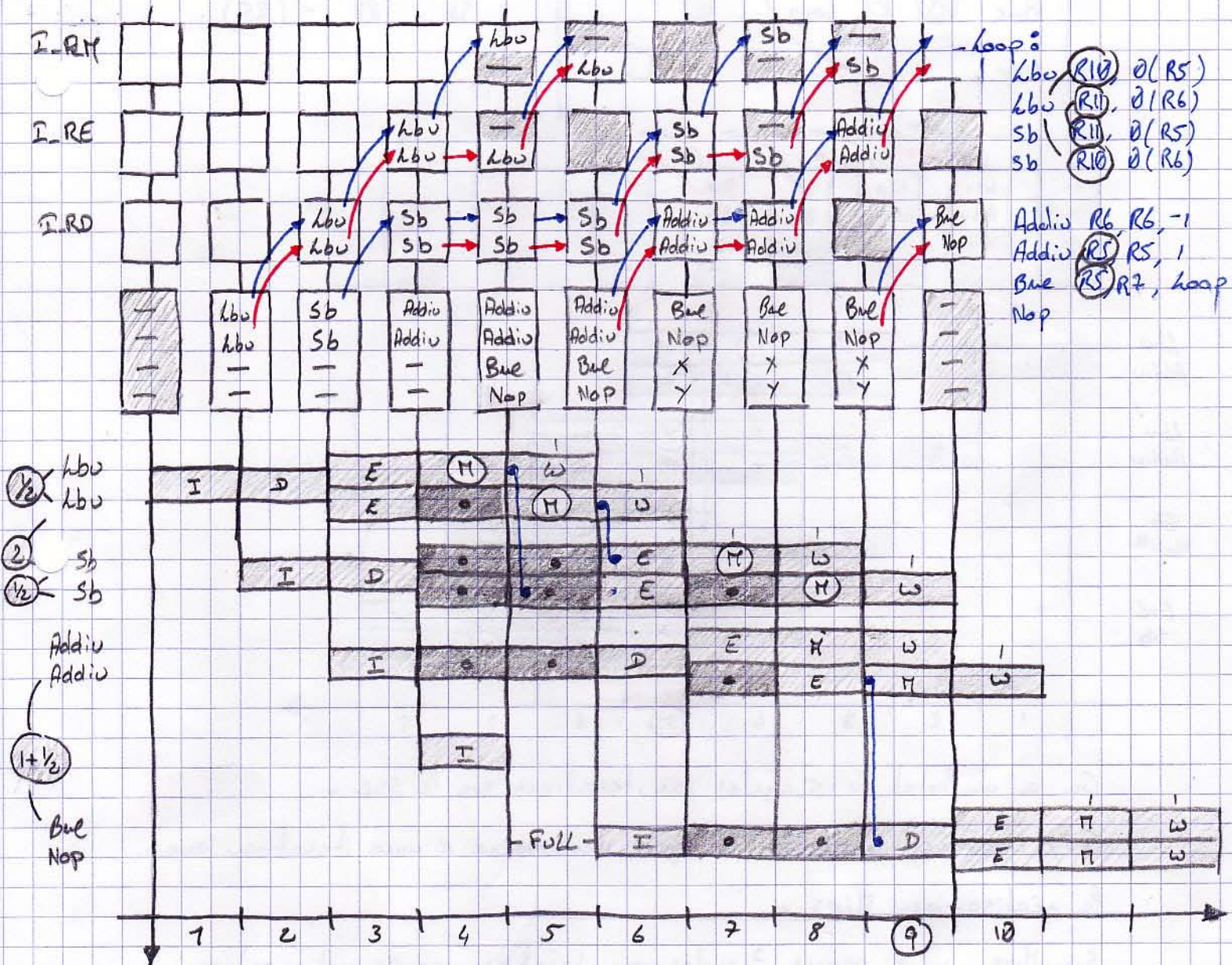
Exe 6 :

Q1/ Modifiez le programme pour qu'il soit exécutable sur SS2.

Il suffit de rajouter un Nop dans le delayed slot du Bne.

Q2/ Analysez, à l'aide d'un schéma simplifié, l'exécution

de ce programme dans le SS2. On suppose qu'au début de l'exécution de la boucle, le buffer d'instructions est vide et que l'étiquette loop est à une adresse alignée, calculez le nb. de cycles par itérations :



On a 9 cycles par itérations sur le SS2.

Q3/ Comparez avec la réalisation MIPS :

On a 10 cycles par itération sur le MIPS.

⇒ Gain = $\frac{10-9}{10} * 100 = 10\%$.

$$\text{Gain}_{\text{SS2/MIPS}} = \frac{\#C_{\text{MIPS}} - \#C_{\text{SS2}}}{\#C_{\text{MIPS}}} * 100$$

Q4/ Optimiser le code par réordonnement. Calculez le nombre de cycles par itérations :

loop :

Lbu R10, 0(R5)

Lbu R11, 0(R6)

Sb R10, 0(R5)

Sb R10, 0(R6)

Addiu R6, R6, -1

Addiu R5, R5, 1

Bne R5, R7, loop

Nop

loop :

Lbu R10, 0(R5)

Addiu R5, R5, 1

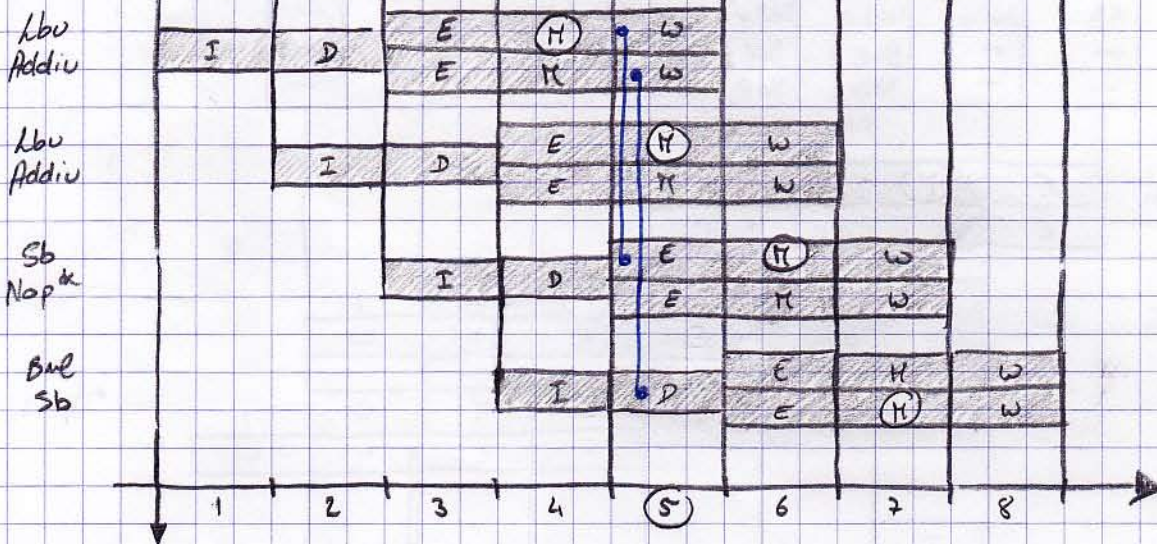
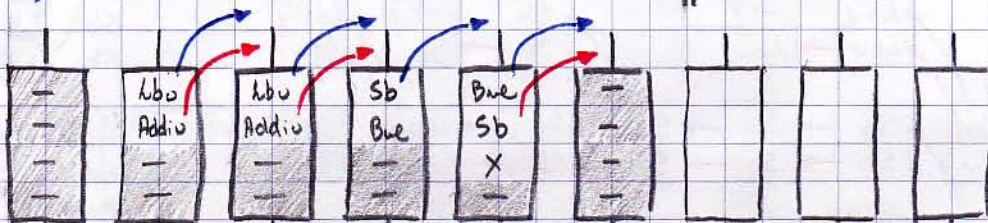
Lbu R11, 0(R6)

Addiu R6, R6, -1

Sb R10, 1(R6)

Bne R5, R7, loop

Sb R11, -1(R5)



On a un total de 5 cycles par itération sur le SSE.

Q5/ Comparer le nombre de cycles nécessaire à une itération avec la réalisation Mips.

Sur Mips, nous avons 7 cycles par itération après optimisation.

On a donc un gain de $\frac{7-5}{7} \times 100 \approx 28\%$.

Exe 6 : Boucles

G6/ Optimiser avec le pipeline logiciel et calculer le nb. de cycles nécessaires à une itération.

Version Initiale :

loop :

```

    Lbu  (R10), 0(R5)      (Ei)
    Lbu  (R11), 0(R6)      R11, R10
    Sb   (R11), 0(R5)      (Ei)
    Sb   (R10), 0(R6)      (Ei)
    Addiu R6, R6, -1
    Addiu R5, R5, 1
    Bne  R7, R5, loop
    Nop
    
```

Version Réorganisée :

loop :

```

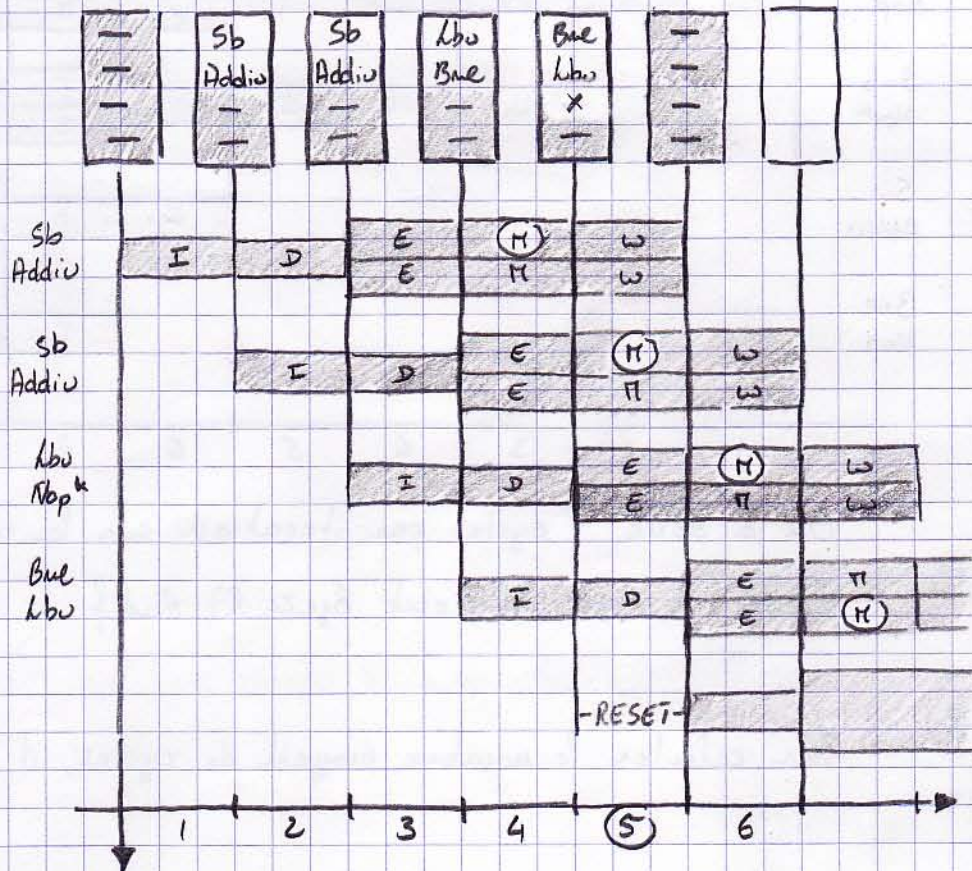
    Sb   R11, -1(R5)      E(i-1)
    Sb   R10, 1(R6)
    Lbu  R10, 0(R5)      E(i)
    Lbu  R11, 0(R6)
    Addiu R6, R6, -1
    Addiu (R5), R5, 1
    Bne  R7, (R5) loop
    Nop
    
```

version Optimisée :

loop :

```

    Sb   R11, -1(R5)
    Addiu R5, R5, 1
    Sb   R10, 1(R6)
    Addiu R6, R6, -1
    Lbu  R11, 1(R6)
    Bne  R5, R7, loop
    Lbu  R10, -1(R5)
    
```



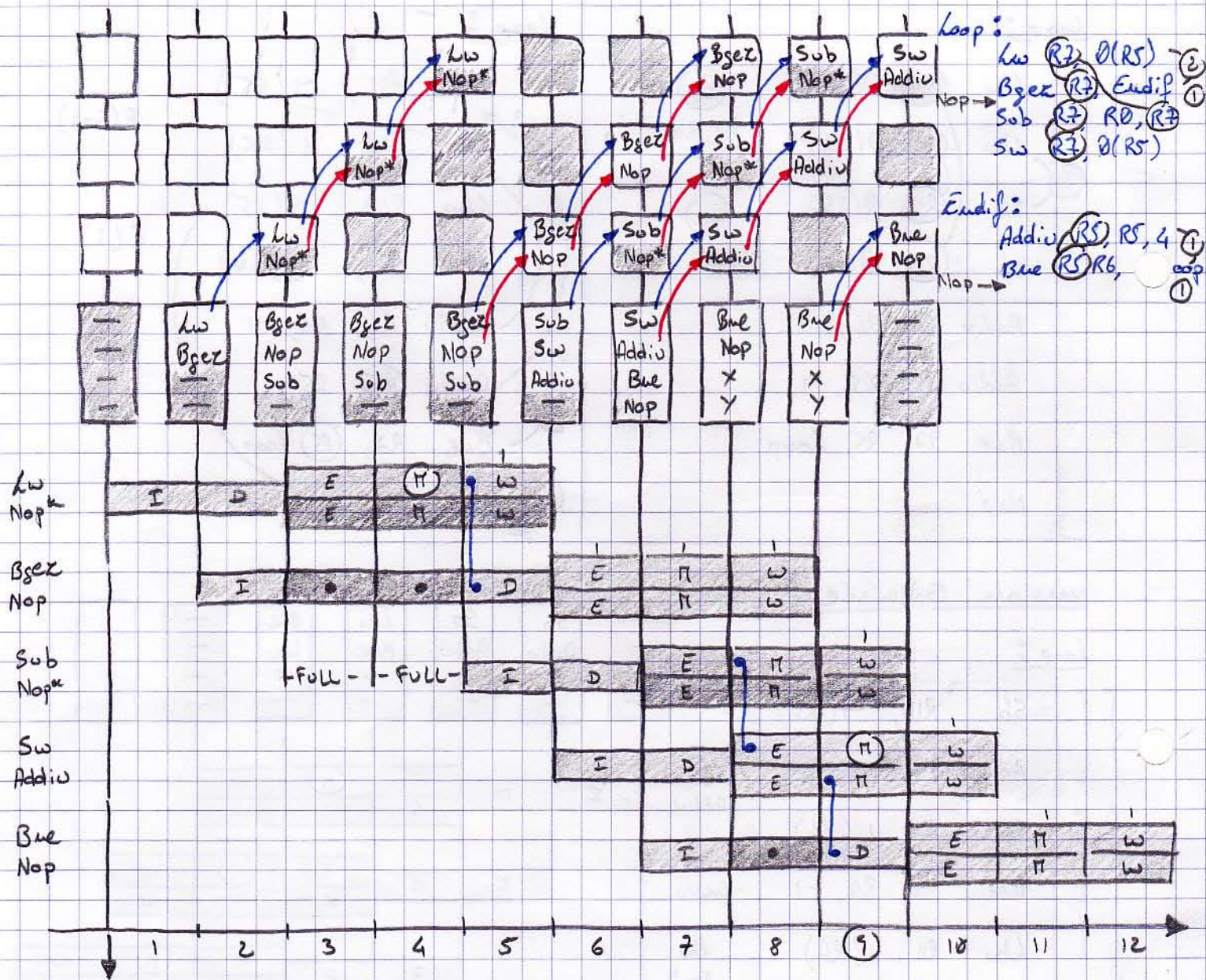
On a donc 5 cy/iter.

Exe 7 :

Q1 & 2/ Modifiez le programme pour qu'il soit exécutable sur le SS2.

Analysez, à l'aide d'un schéma simplifié, l'exécution de la boucle dans le SS2. On suppose qu'au début de l'exécution, le buffer d'instructions est vide et que l'étiquette loop se trouve à une adresse alignée.

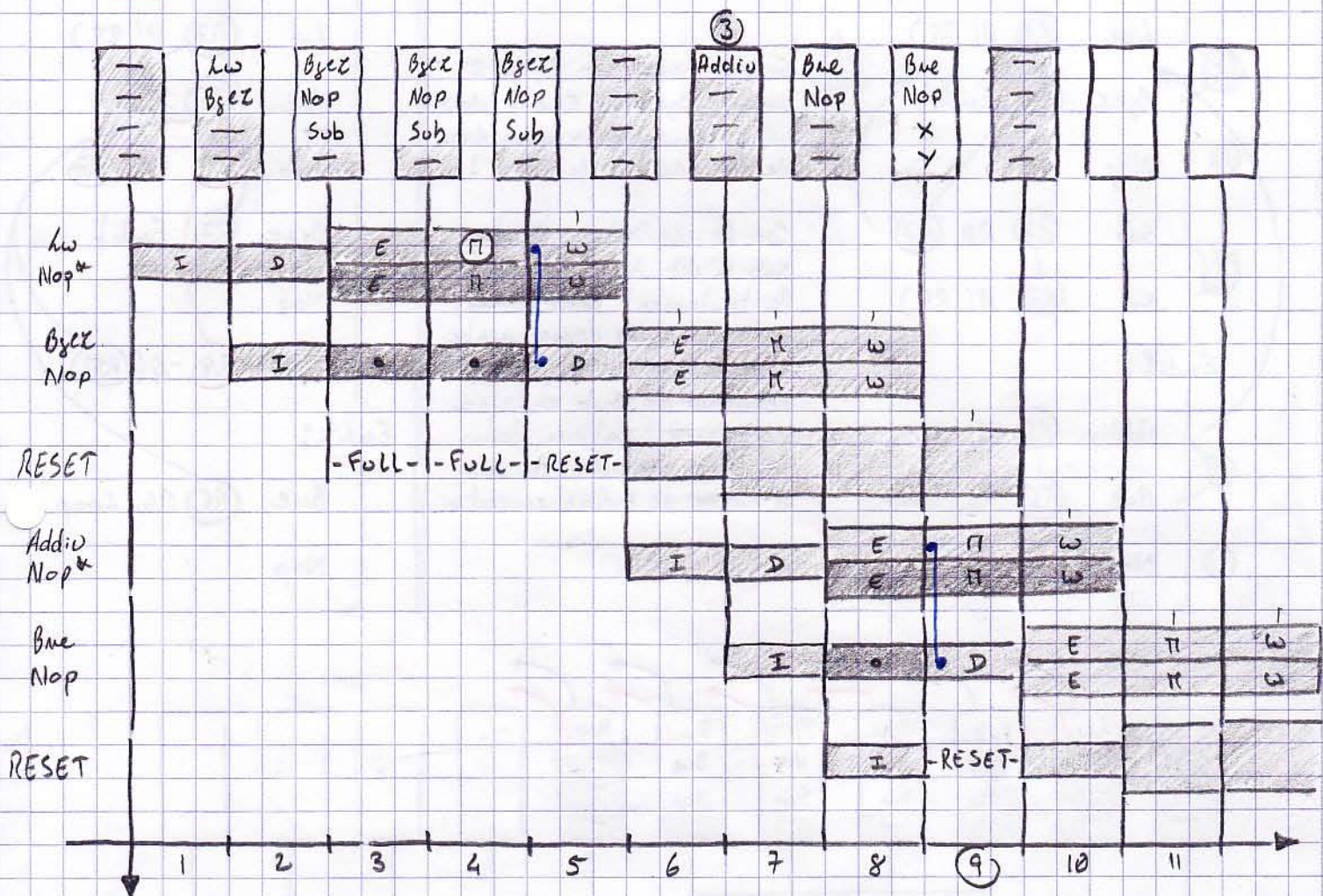
Calculez le nombre moyen de cycles par itérations :



On a donc 9 cycles par itération sur la réalisation SS2 en cas de l'échec du branchement Bgez R7, Endif.

Pour calculer le nombre moyen de cycles, il faut faire un 2nd schéma.

Exe 7 : Suite



On a donc également 9 cycles par itération sur le SSE en cas de réussite du branchement Bgez RT, Endif.

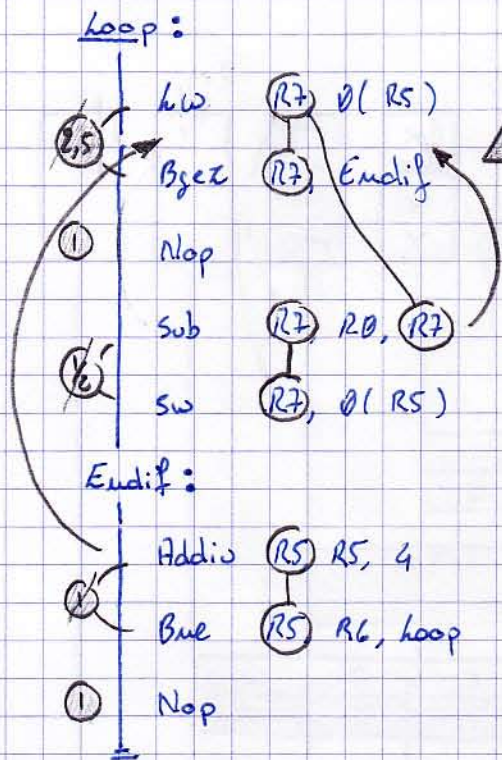
Donc nous avons en moyenne 9 cycles par itération.

Q3/ Comparer le nombre moyen de cycles par itération avec la réalisation Tips.

Sur une réalisation Tips, nous avons eu cas d'échec 11 cy/iten et en cas de Réussite 9 cy/iten. Concernant les probabilités rien n'a été spécifié donc nous supposons que le branchement échoue dans 50% des cas, on a : $\tilde{C}_{Tips} = 9 * 0.5 + 11 * 0.5 = 10 \text{ cy/iten}$.

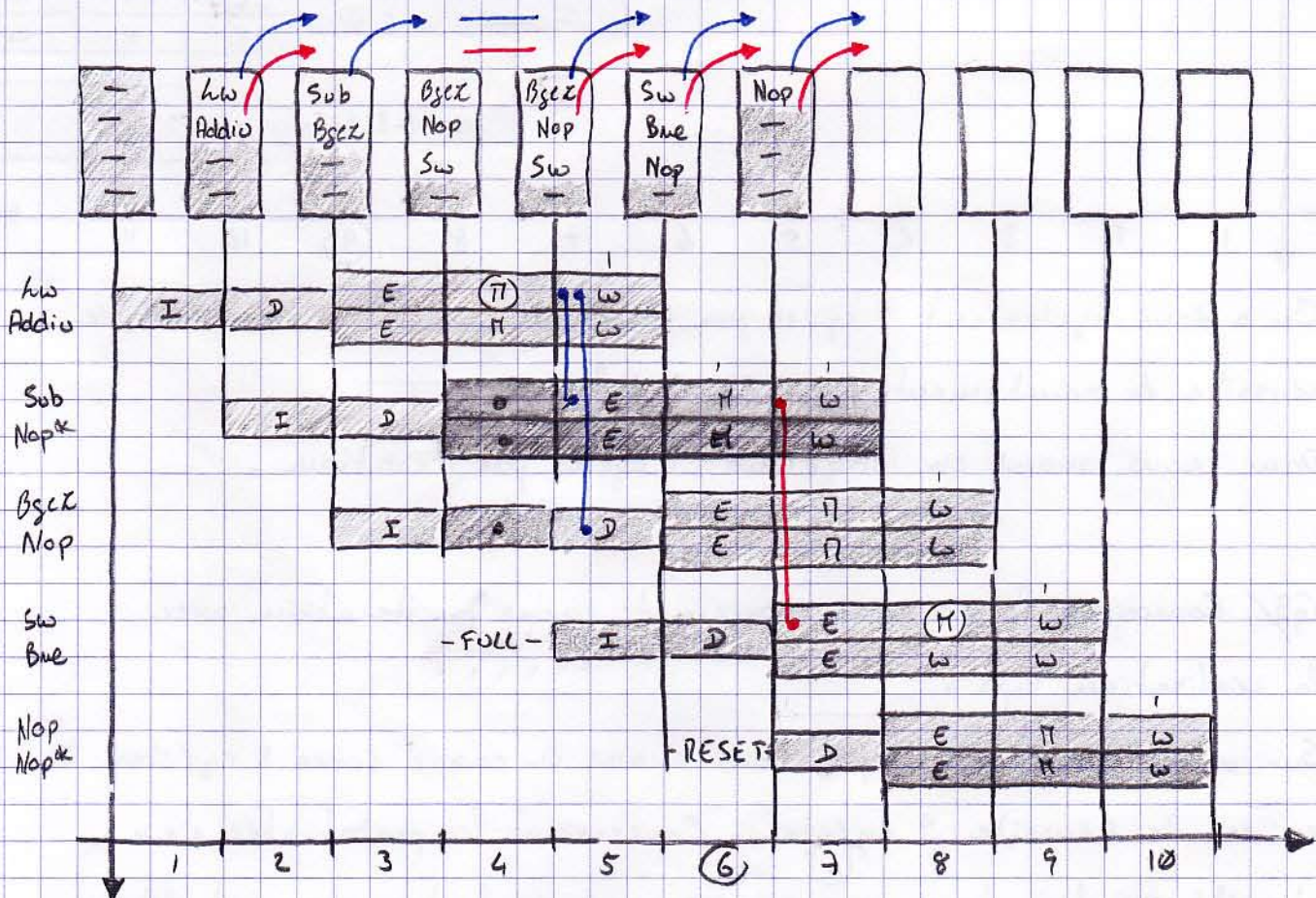
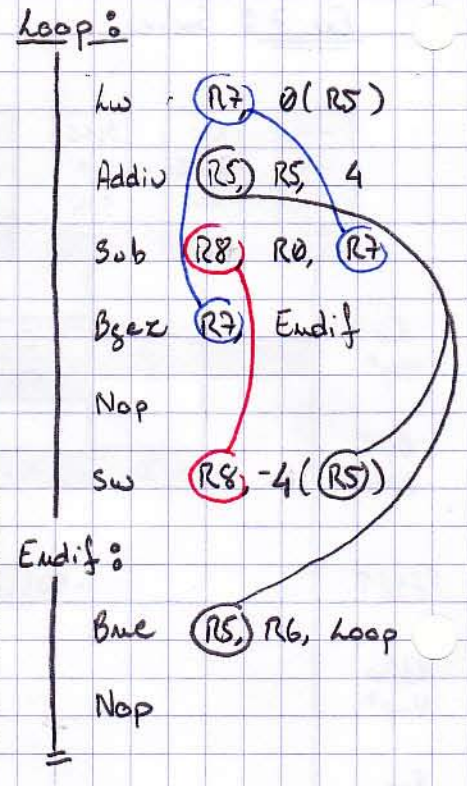
Le gain est donc de $\frac{10-9}{10} * 100 = 10\%$.

64/ Optimisez le code par réordonnement. Calculez le nombre de cycles par itération.

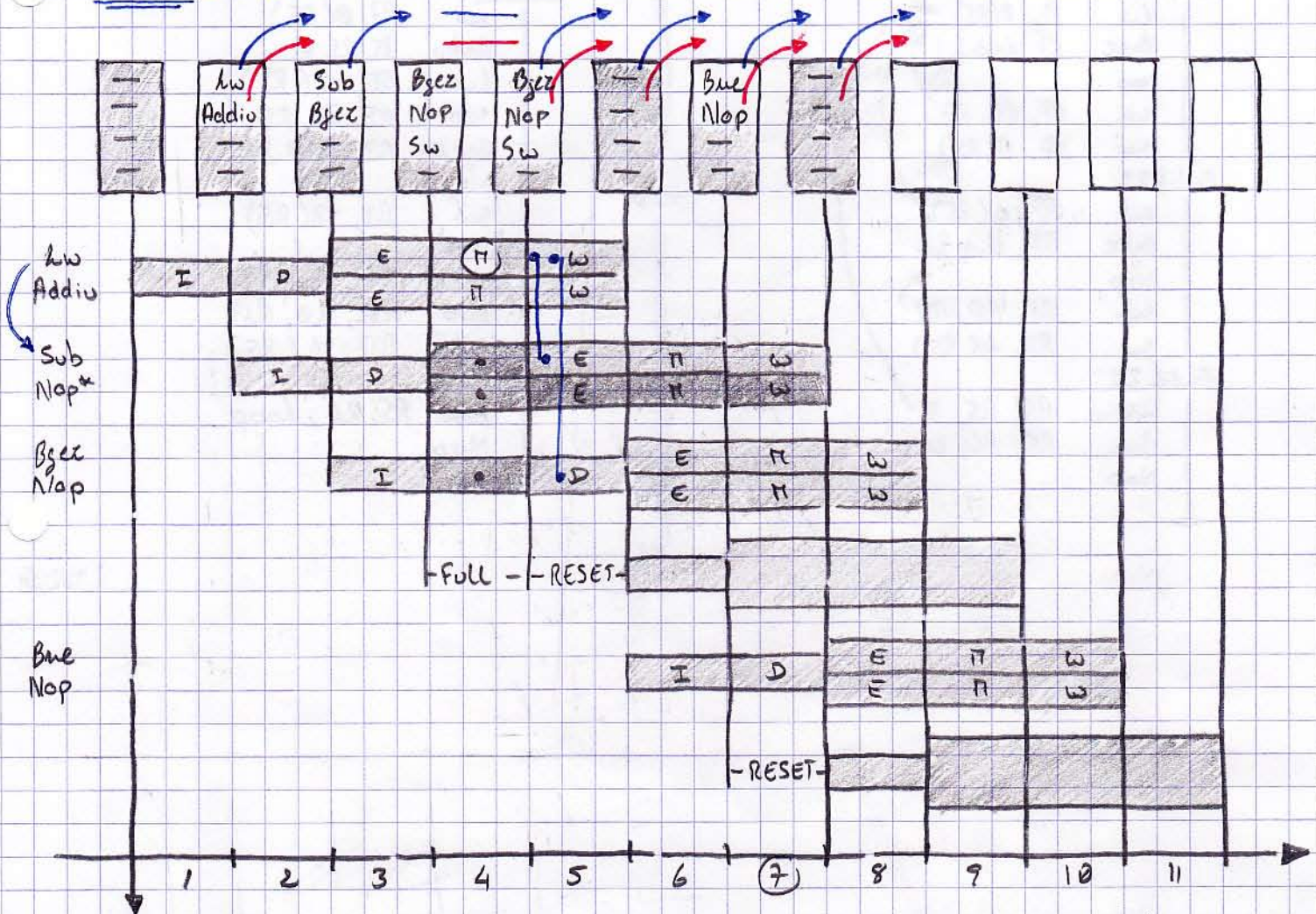


Si on place `sub R7, R0, R7` avant `Bgez R7, Endif` alors nous causerions une dep. de donnée sur le `Bgez` !

Or le résultat de l'instr. `sub` n'est utilisé que dans le traitement conditionnel, nous pouvons donc juste renommer le registre du résultat de `sub` et faire cette modification dans le bloc exécuté en cas d'échec du branchement.



On a donc 6 cycles par itération sur la réalisation SSZ dans le cas de l'échec du branchement.

Exe 7 : Suite

Nous avons donc 7 cycles par itération sur le SS2 dans le cas d'une réussite du branchement.

Donc $\widetilde{C}_{SS2} = 6 * 0,5 + 7 * 0,5 = 6,5 \text{ cy/iten.}$

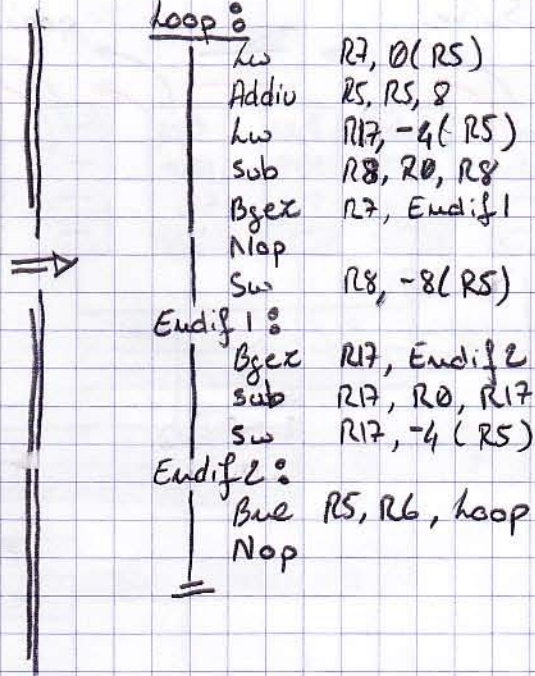
Q5/ Comparez avec la réalisation Pips :

Sur Pips : $\left. \begin{array}{l} \#C_{Ech} = 9 \text{ cy/iten.} \\ \#C_{Réu} = 8 \text{ cy/iten.} \end{array} \right\} \Rightarrow \widetilde{C}_{Pips} = 9 * 0,5 + 8 * 0,5 = 8,5 \text{ cy/iten.}$

Sur SS2 : $\left. \begin{array}{l} \#C_{Ech} = 6 \text{ cy/iten.} \\ \#C_{Réu} = 7 \text{ cy/iten.} \end{array} \right\} \Rightarrow \widetilde{C}_{SS2} = 6 * 0,5 + 7 * 0,5 = 6,5 \text{ cy/iten.}$

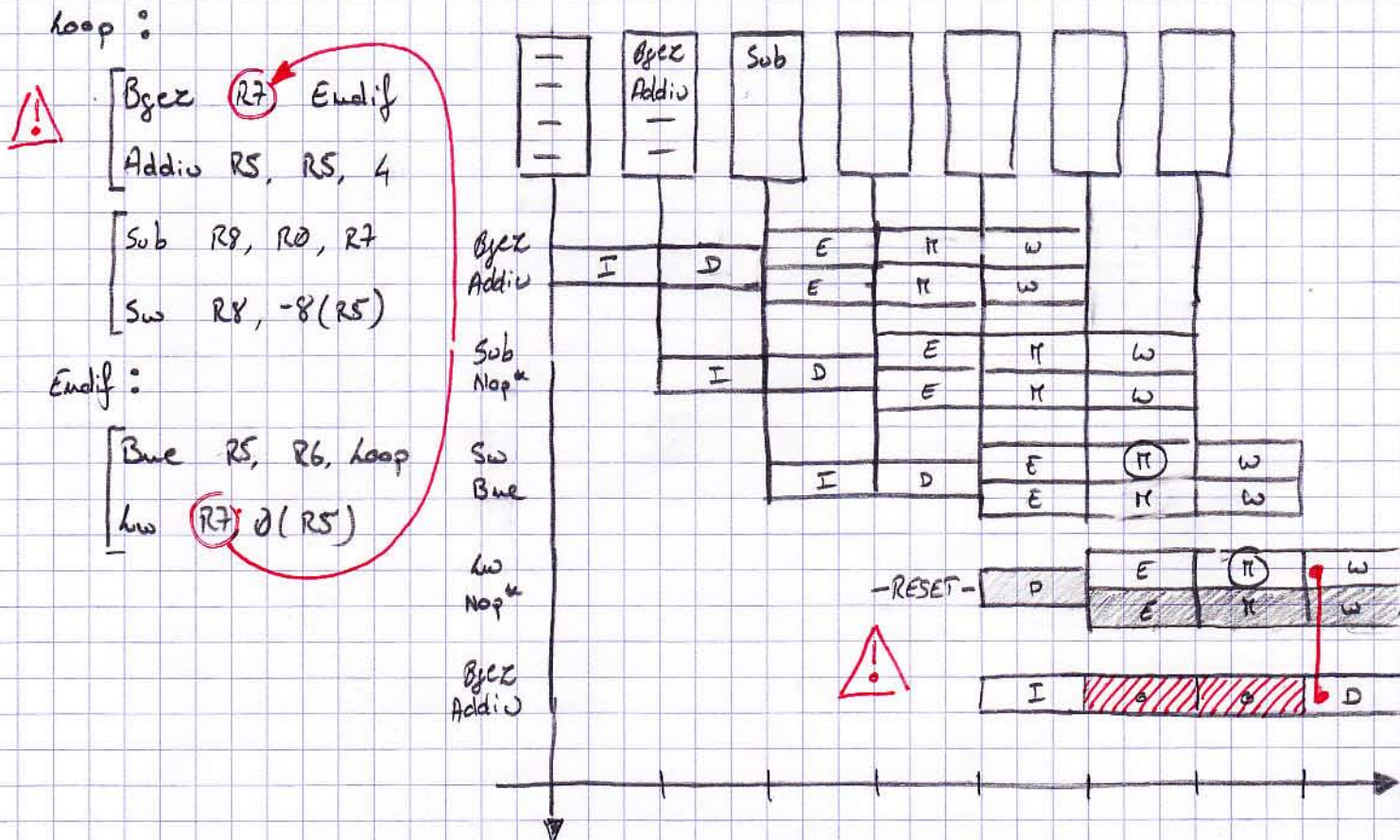
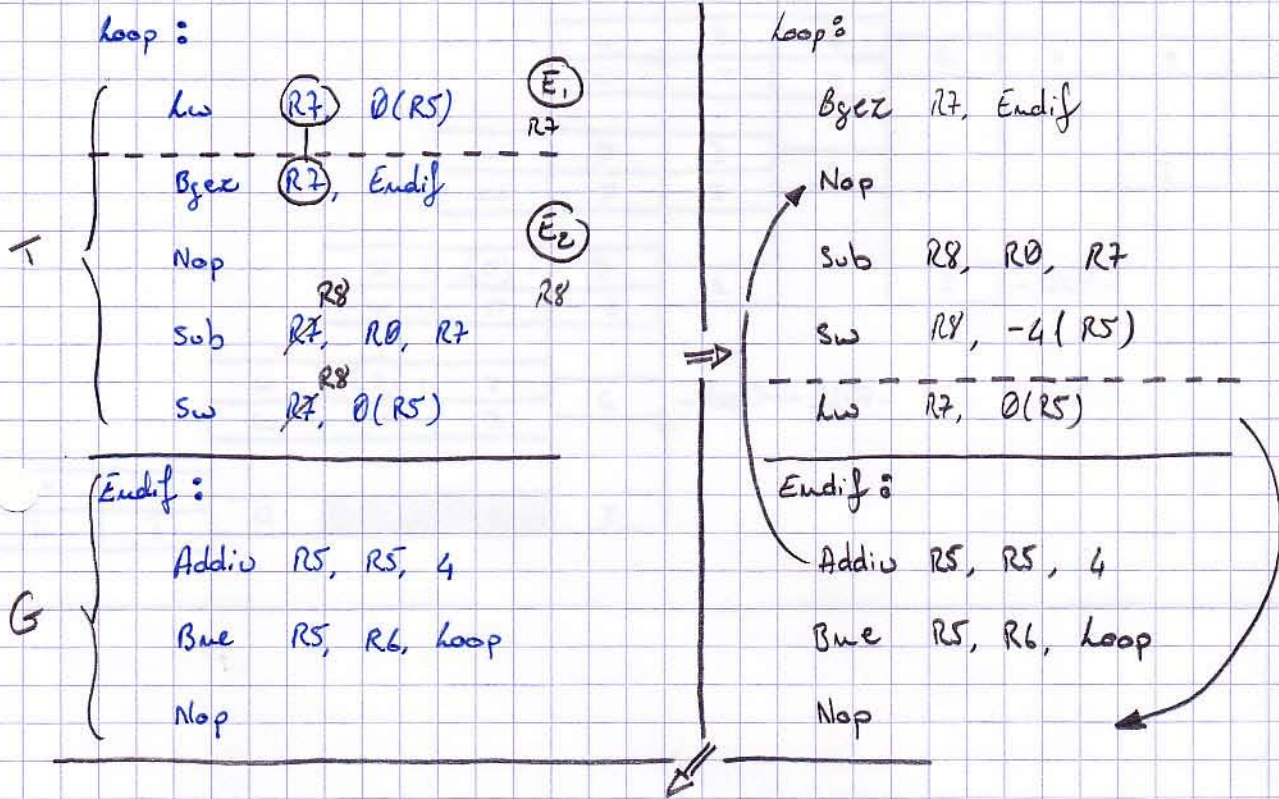
On a donc un gain de $\frac{8,5 - 6,5}{8,5} * 100 \approx 23,5\%$.

Q6/ Déroulez la boucle 1 fois (traitement de deux éléments par iteration)
et optimiser . Calculez le nombre de cycles par iteration .

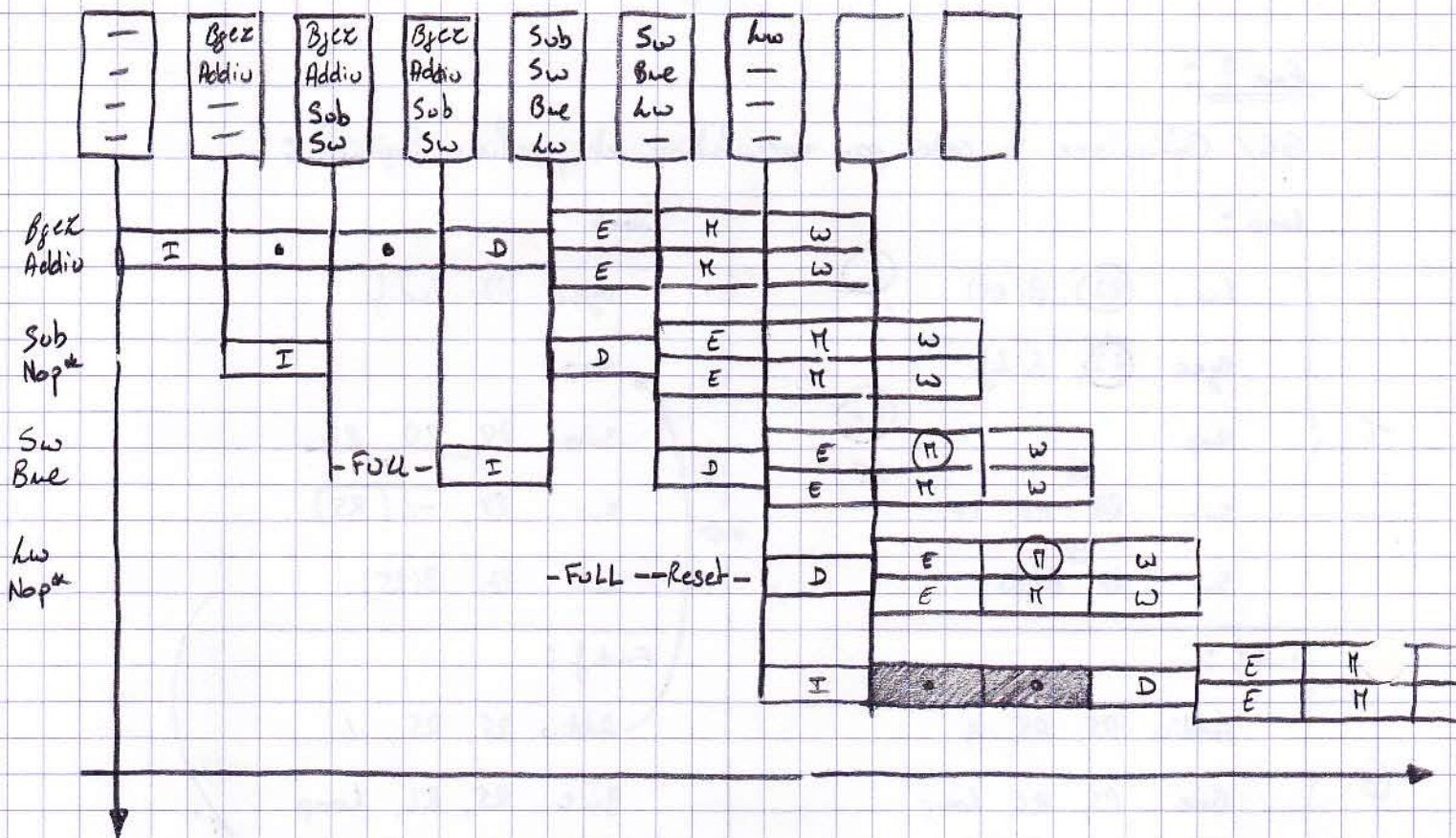


Exe 7 :

86/ Optimisez le code par speculation et pipeline logiciel :



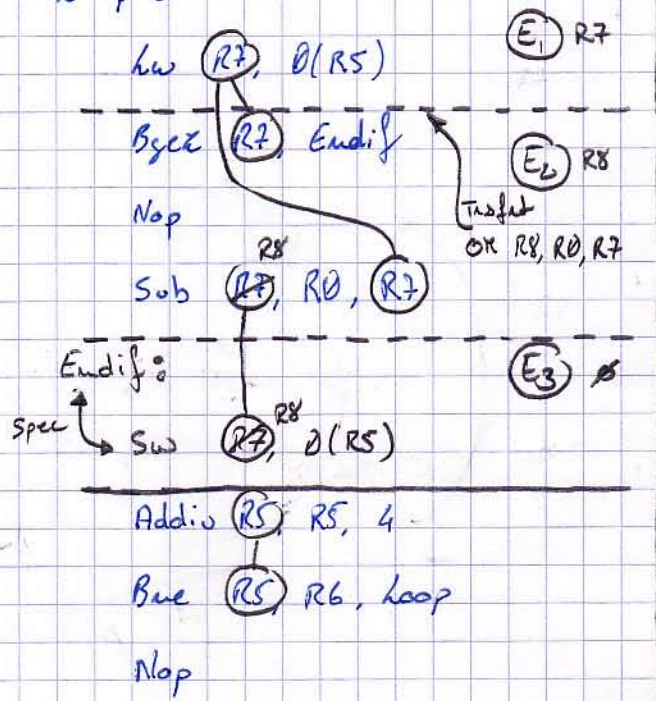
On a donc un nouveau schéma :



Exe 7 :

Q7/ Optimisez le code par speculation puis par pipeline logiciel.

loop :



loop :

