

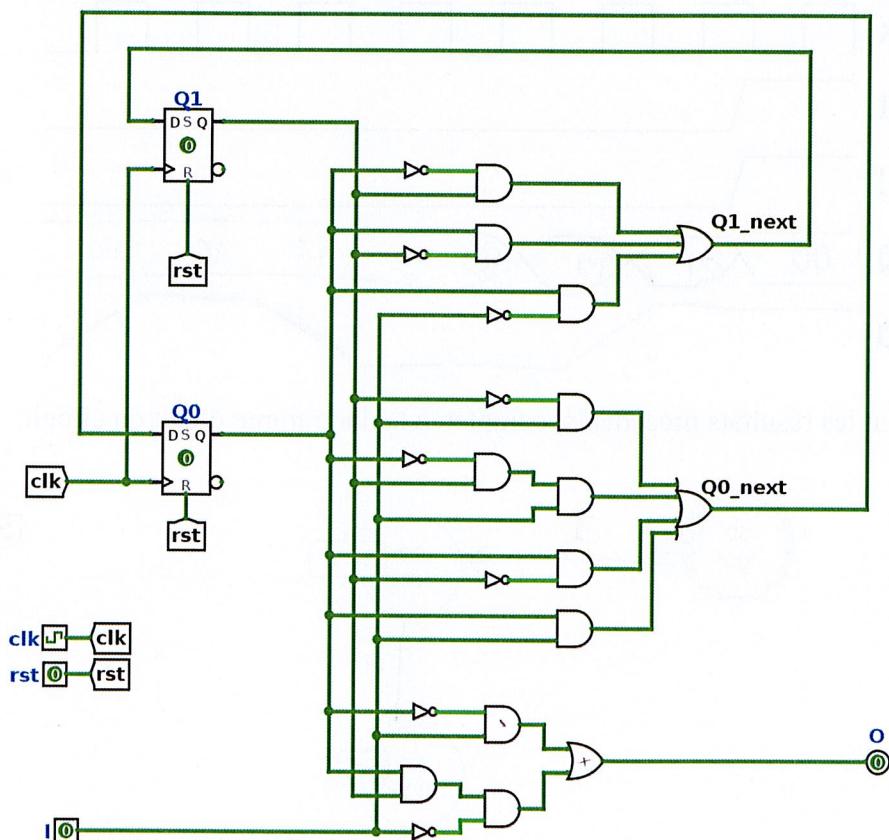
- Durée de l'épreuve : 2 heures
- Polycopié de cours autorisé, **autres documents interdits**, calculatrices non connectées autorisées

|            |  |
|------------|--|
| Exercice 1 |  |
| Exercice 2 |  |
| Total      |  |

**Attention, toutes les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.**

### Exercice 1 : machine à états (7 points)

On s'intéresse à un circuit séquentiel à une entrée  $I$  et une sortie  $O$ , et dont l'état interne est codé par un registre 2 bits  $Q[1:0]$ . Le registre dispose d'une commande  $rst$  (remise à zéro) qui est utilisée pour forcer l'état initial de la machine à 00.



**Q1 :** S'agit-il d'une machine de Mealy ou de Moore ? Justifiez votre réponse.

C'est une machine de Mealy car la sortie  $O$  dépend l'entrée  $I$   
es ~~échappent~~ des registres  $Q_0$  et  $Q_1$  (basades  $Q_0$  et  $Q_1$ )



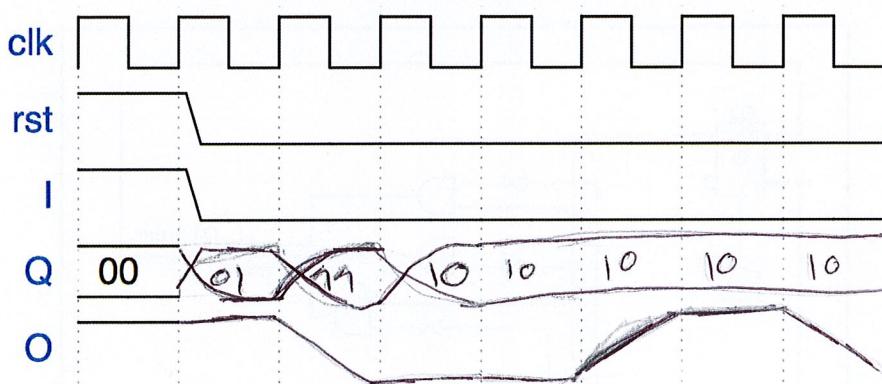
**Q3 :** Donner l'expression booléennes associée à  $Q_1^{t+1}$ .

$$Q_1^{t+1} = (\bar{Q}_0^t \cdot Q_1^t) + (Q_0^t \cdot \bar{Q}_1^t) + (Q_0^t \cdot I)$$

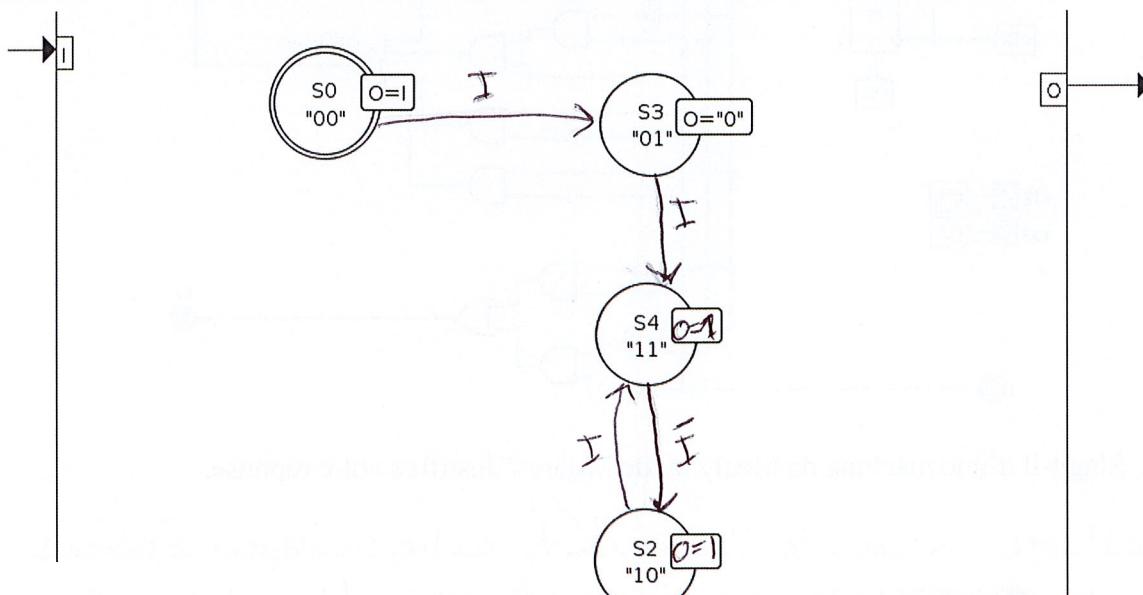
**Q4 :** Compléter la table de transition de la machine à état correspondant à ce circuit.

|   | $Q_1^t$ | $Q_0^t$ | $I^t$ | $Q_1^{t+1}$ | $Q_0^{t+1}$ | $O^t$ |
|---|---------|---------|-------|-------------|-------------|-------|
| 0 | 0       | 0       | 0     | 0           | 0           | 0     |
| 1 | 0       | 0       | 1     | 0           | 1           | 1     |
| 2 | 0       | 1       | 0     | 1           | 1           | 0     |
| 3 | 0       | 1       | 1     | 1           | 1           | 0     |
| 4 | 1       | 0       | 0     | 1           | 0           | 0     |
| 5 | 1       | 0       | 1     | 1           | 1           | 1     |
| 6 | 1       | 1       | 0     | 1           | 0           | 1     |
| 7 | 1       | 1       | 1     | 0           | 1           | 0     |

**Q5 :** Complétez le chronogramme avec les valeurs de Q et O.



**Q6 :** En utilisant les résultats précédents, complétez le diagramme d'état du circuit.



### Exercice 3 : décomposition UT/UC (13 points)

On souhaite réaliser un circuit permettant de calculer le produit de deux nombres complexes  $a$  et  $b$ , dont les parties réelles et imaginaires sont représentées par des entiers sur 16 bits. Le produit est calculé selon l'algorithme ci-dessous :

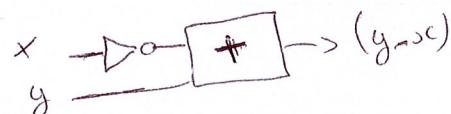
```
debut
    a_re := lire_entree();
    a_im := lire_entree();
    b_re := lire_entree();
    b_im := lire_entree();
    c_re := a_re.b_re-a_im.b_im;
    c_im := a_re.b_im+a_im.b_re;
    écrire_sortie(c_re);
    écrire_sortie(c_im);
fin;
```

Afin de réduire le nombre de portes logiques utilisées par le circuit, on fait le choix de limiter le nombre d'opérateurs arithmétique en n'utilisant qu'un seul additionneur et un seul multiplicateur, comme indiqué sur le circuit de la question 4.

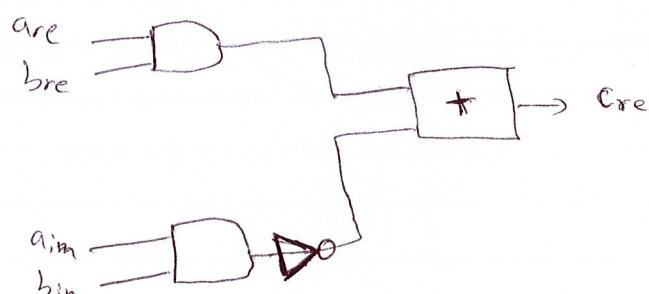
**Question 1 :** L'algorithme requiert des opérations de soustraction, or notre composant ne dispose que d'additionneurs. Expliquez (en vous aidant d'un schéma) comment il est possible de réaliser un circuit soustracteur à partir d'un additionneur et d'un ensemble de portes « inverseur ».

Il est possible de rajouter une porte logique pour la négation

en général:



sur notre exemple:



**Question 2 :** Le circuit de la question 4 permet de réaliser une addition et une multiplication en parallèle dans un même cycle. Utilisez cette propriété pour réécrire l'algorithme de multiplication de façon à minimiser le nombre d'étapes (cycles) de calcul. Vous utiliserez pour cela les conventions vues en cours et rappelées dans le tableau ci-après, et pourrez introduire des variables intermédiaires afin de faciliter la compréhension du code.

| Exécution    | Notation   | Signification  |
|--------------|--|--|
| Parallèle    | $\mathbf{X} = \mathbf{Y} + 1$ , $\mathbf{Y} = \mathbf{X} - 1$ ;    | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ et $\mathbf{Y}^{t+1} = \mathbf{X}^t - 1$ ;           |
| Séquentielle | $\mathbf{X} = \mathbf{Y} + 1$ ;<br>$\mathbf{Y} = \mathbf{X} - 1$ ; | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ suivi de $\mathbf{Y}^{t+2} = \mathbf{X}^{t+1} - 1$ ; |

```

début
    are := lire_entree();
    aim := lire_entree();
    bre := lire_entree();
    bim := lire_entree();

```

$$1 \quad \cancel{x^{t+1}} = y^t + 1, \cancel{y^{t+1}} = x^t - 1$$

$$2 \quad \cancel{y^{t+1}} = x^t - 1;$$

$$3 \quad \cancel{x^{t+1}} = y^t + 1;$$

4

5

6

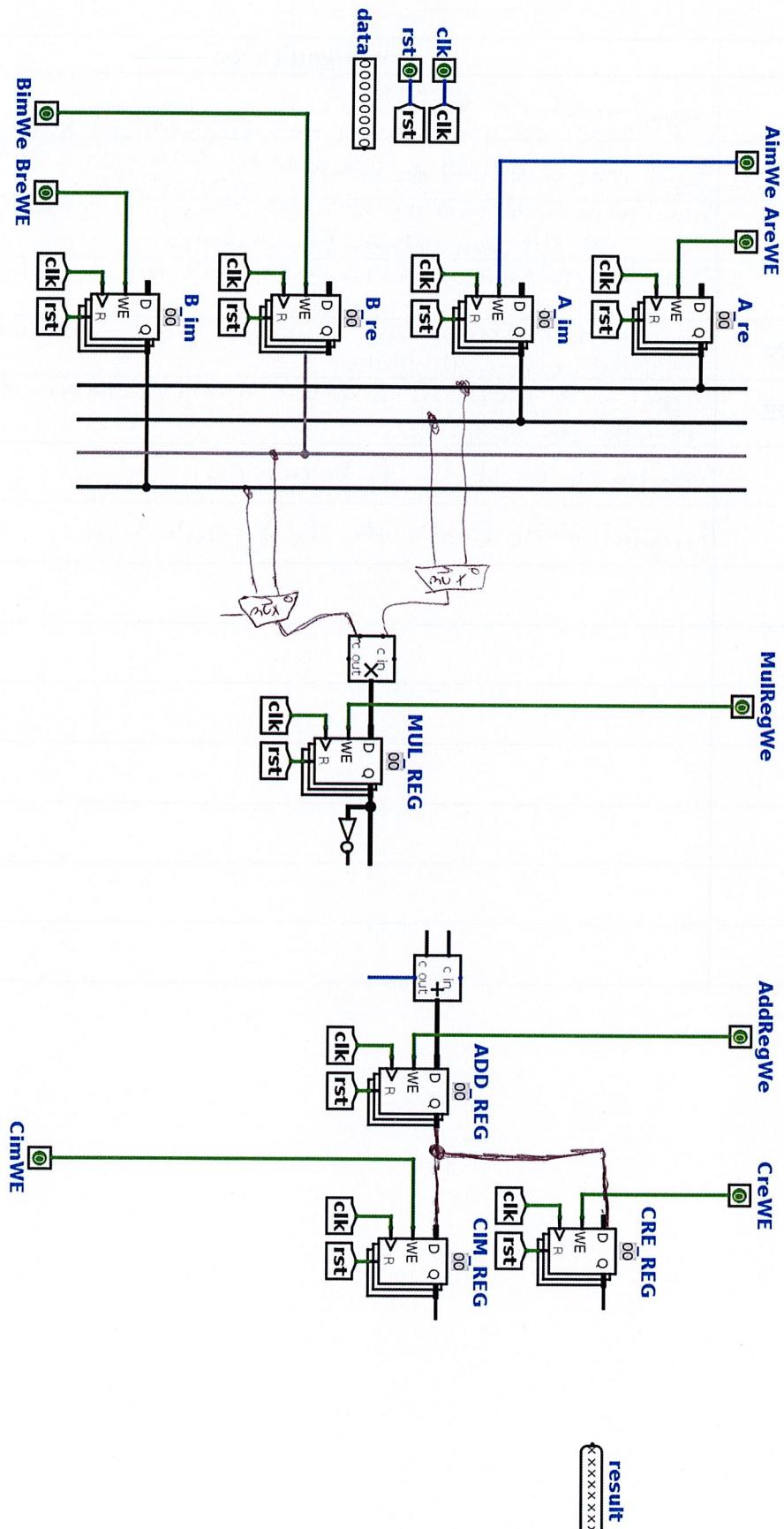
**fin;**

**Question 3 :** sachant que les entrées sorties se font sur le principe du protocole req/ack, combien de cycles faut-il au minimum pour réaliser l'ensemble des traitements (calculs et entrées/sorties) ? Veillez à bien justifier vos réponses

6 cycles



**Question 4 :** Utilisez les résultats des questions précédentes pour compléter le schéma de l'unité de traitement du circuit, en y ajoutant les interconnexions, multiplexeurs, etc. permettant de réaliser le traitement selon l'approche proposée aux questions 1 et 2.

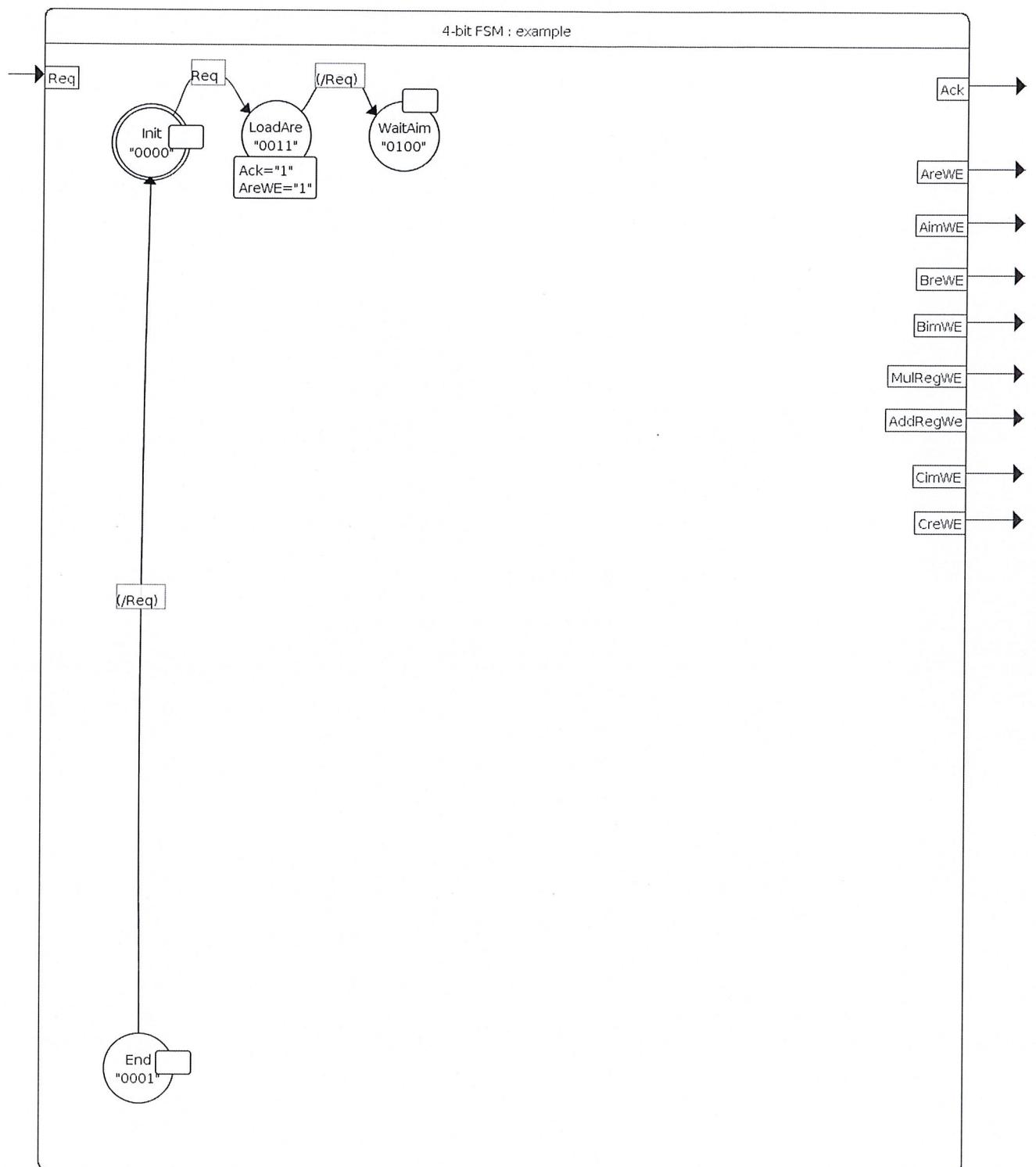


**Question 5 :** Complétez le tableau ci-dessous avec les signaux de contrôle supplémentaires, en explicitant à chaque fois leur rôle .

| Nom             | Rôle/signification  |
|-----------------|---|
| <b>AimWE</b>    | Signal write Enable de la bascule Aim, si elle est à 1 la bascule n'incremente pas (grâce au signal de top d'horloge) |
| <b>AreWE</b>    | Signal write Enable (WE) de la bascule Are, si le signal est à 0 elle sera activée                                    |
| <b>BimWE</b>    | Signal write Enable (WE) de la bascule Bim, Si le signal est à 0 elle sera activée (fonctionnelle)                    |
| <b>BreWE</b>    | Signal write Enable (WE) de la bascule Bre, Si le signal est à 0 elle est activée <del>sur</del>                      |
| <b>MulRegWE</b> | Signal write Enable (WE) de la bascule mulReg qui recevra le résultat de la multiplication                            |
| <b>AddRegWE</b> | Signal write Enable de la bascule ADD_Reg qui recevra le résultat de l'addition                                       |
| <b>CreWE</b>    | Signal write Enable de la bascule Cre   |
| <b>CinWE</b>    | Signal write Enable de la bascule Cin   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |



**Question 6 :** complétez diagramme de transition de l'unité de contrôle du circuit, en incluant la gestion des entrées/sorties.



http://www.jingyan.com/jingyan/



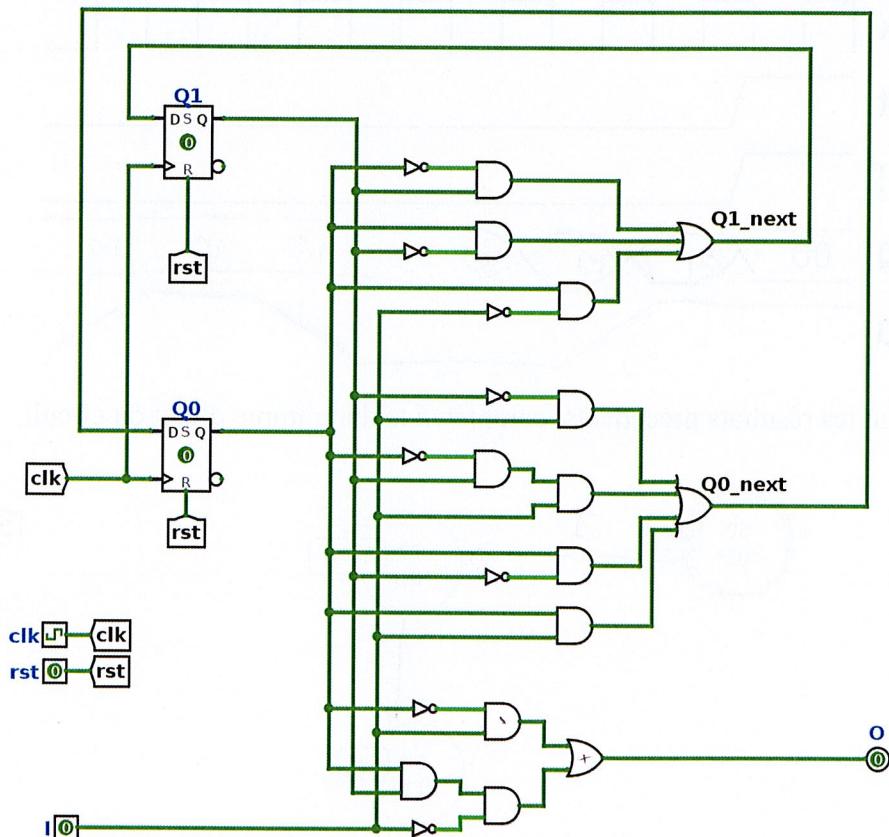
- Durée de l'épreuve : 2 heures
- Polycopié de cours autorisé, **autres documents interdits**, calculatrices non connectées autorisées

|            |  |
|------------|--|
| Exercice 1 |  |
| Exercice 2 |  |
| Total      |  |

**Attention, toutes les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.**

### Exercice 1 : machine à états (7 points)

On s'intéresse à un circuit séquentiel à une entrée  $I$  et une sortie  $O$ , et dont l'état interne est codé par un registre 2 bits  $Q[1:0]$ . Le registre dispose d'une commande  $rst$  (remise à zéro) qui est utilisée pour forcer l'état initial de la machine à 00.



**Q1 :** S'agit-il d'une machine de Mealy ou de Moore ? Justifiez votre réponse.

C'est une machine de Mealy car la sortie  $O$  dépend de l'entrée  $I$   
et des états ~~initials~~ des registres  $Q_0$  et  $Q_1$  (basades  $Q_0$  et  $Q_1$ )



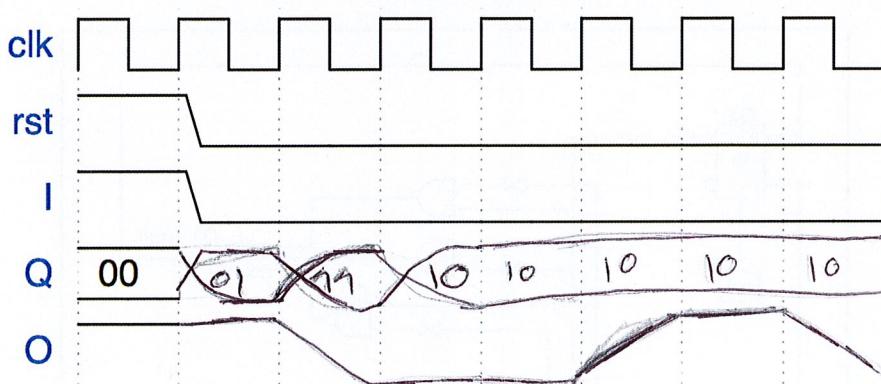
**Q3 :** Donner l'expression booléennes associée à  $Q_1^{t+1}$ .

$$Q_1^{t+1} = (\bar{Q}_0^t \cdot Q_1^t) + (Q_0^t \cdot \bar{Q}_1^t) + (Q_0^t \cdot I)$$

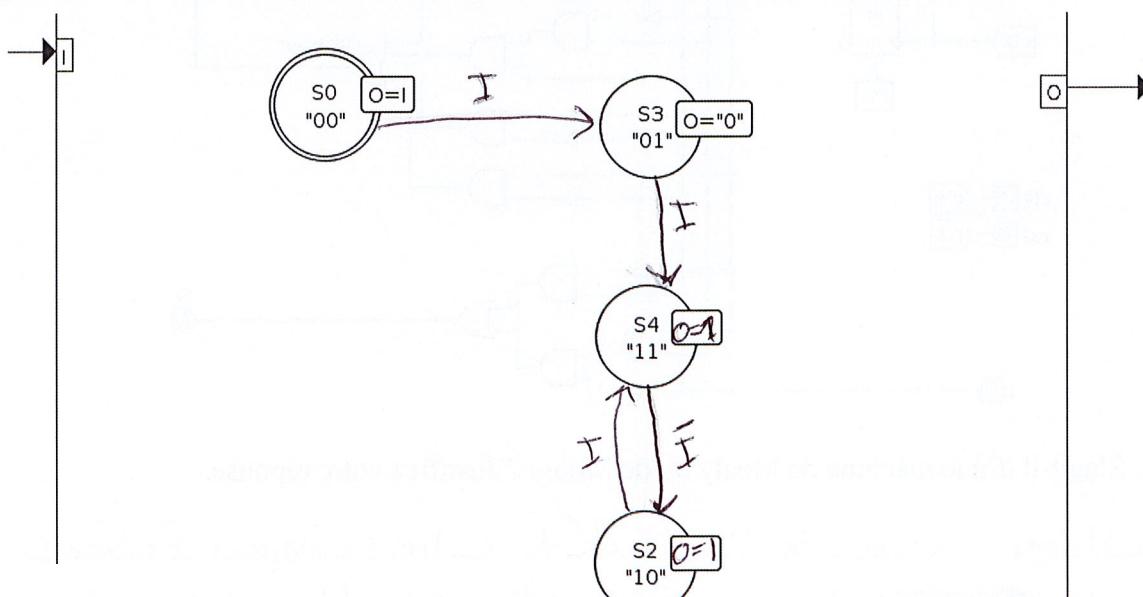
**Q4 :** Compléter la table de transition de la machine à état correspondant à ce circuit.

|   | $Q_1^t$ | $Q_0^t$ | $I^t$ | $Q_1^{t+1}$ | $Q_0^{t+1}$ | $O^t$ |
|---|---------|---------|-------|-------------|-------------|-------|
| 0 | 0       | 0       | 0     | 0           | 0           | 0     |
| 1 | 0       | 0       | 1     | 0           | 1           | 1     |
| 2 | 0       | 1       | 0     | 1           | 1           | 0     |
| 3 | 0       | 1       | 1     | 1           | 1           | 0     |
| 4 | 1       | 0       | 0     | 1           | 0           | 0     |
| 5 | 1       | 0       | 1     | 1           | 1           | 1     |
| 6 | 1       | 1       | 0     | 1           | 0           | 1     |
| 7 | 1       | 1       | 1     | 0           | 1           | 0     |

**Q5 :** Complétez le chronogramme avec les valeurs de Q et O.



**Q6 :** En utilisant les résultats précédents, complétez le diagramme d'état du circuit.



### Exercice 3 : décomposition UT/UC (13 points)

On souhaite réaliser un circuit permettant de calculer le produit de deux nombres complexes  $a$  et  $b$ , dont les parties réelles et imaginaires sont représentées par des entiers sur 16 bits. Le produit est calculé selon l'algorithme ci-dessous :

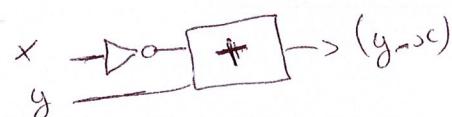
```
debut
    a_re := lire_entree();
    a_im := lire_entree();
    b_re := lire_entree();
    b_im := lire_entree();
    c_re := a_re.b_re-a_im.b_im;
    c_im := a_re.b_im+a_im.b_re;
    écrire_sortie(c_re);
    écrire_sortie(c_im);
fin;
```

Afin de réduire le nombre de portes logiques utilisées par le circuit, on fait le choix de limiter le nombre d'opérateurs arithmétique en n'utilisant qu'un seul additionneur et un seul multiplicateur, comme indiqué sur le circuit de la question 4.

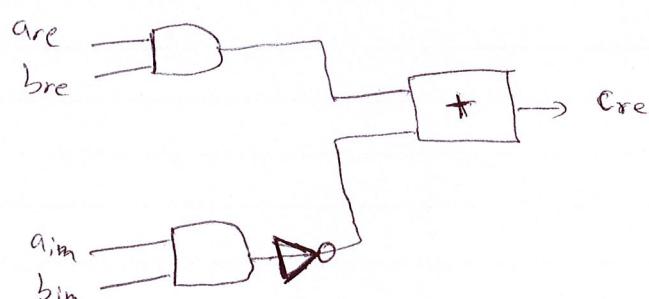
**Question 1 :** L'algorithme requiert des opérations de soustraction, or notre composant ne dispose que d'additionneurs. Expliquez (en vous aidant d'un schéma) comment il est possible de réaliser un circuit soustracteur à partir d'un additionneur et d'un ensemble de portes « inverseur ».

Il est possible de rajouter une porte logique pour la négation

en général:



sur notre exemple:



**Question 2 :** Le circuit de la question 4 permet de réaliser une addition et une multiplication en parallèle dans un même cycle. Utilisez cette propriété pour réécrire l'algorithme de multiplication de façon à minimiser le nombre d'étapes (cycles) de calcul. Vous utiliserez pour cela les conventions vues en cours et rappelées dans le tableau ci-après, et pourrez introduire des variables intermédiaires afin de faciliter la compréhension du code.

| Exécution    | Notation   | Signification  |
|--------------|--|--|
| Parallèle    | $\mathbf{X} = \mathbf{Y} + 1$ , $\mathbf{Y} = \mathbf{X} - 1$ ;    | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ et $\mathbf{Y}^{t+1} = \mathbf{X}^t - 1$ ;           |
| Séquentielle | $\mathbf{X} = \mathbf{Y} + 1$ ;<br>$\mathbf{Y} = \mathbf{X} - 1$ ; | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ suivi de $\mathbf{Y}^{t+2} = \mathbf{X}^{t+1} - 1$ ; |

```

début
    are := lire_entree();
    aim := lire_entree();
    bre := lire_entree();
    bim := lire_entree();

```

$$1 \quad \cancel{x^{t+1}} = y^t + 1, \cancel{y^{t+1}} = x^t - 1$$

$$2 \quad \cancel{y^{t+1}} = x^t - 1;$$

$$3 \quad \cancel{x^{t+1}} = y^t + 1;$$

4

5

6

**fin;**

**Question 3 :** sachant que les entrées sorties se font sur le principe du protocole req/ack, combien de cycles faut-il au minimum pour réaliser l'ensemble des traitements (calculs et entrées/sorties) ? Veillez à bien justifier vos réponses

6 cycles

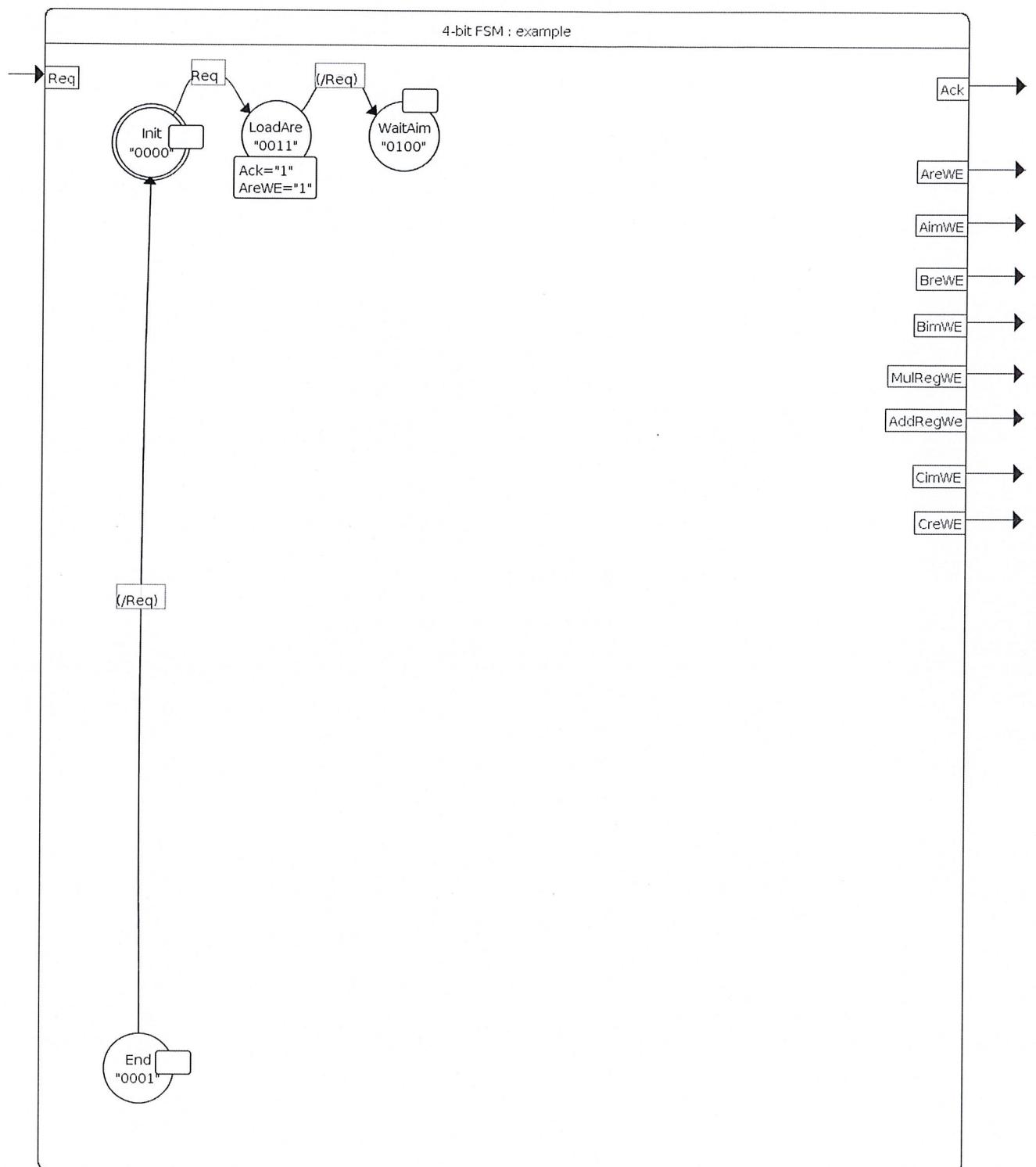


**Question 5 :** Complétez le tableau ci-dessous avec les signaux de contrôle supplémentaires, en explicitant à chaque fois leur rôle .

| Nom             | Rôle/signification  |
|-----------------|---|
| <b>AimWE</b>    | Signal write Enable de la bascule Aim, si elle est à 1 la bascule n'incremente pas (grâce au signal de top d'horloge) |
| <b>AreWE</b>    | Signal write Enable (WE) de la bascule Are, si le signal est à 0 elle sera activée                                    |
| <b>BimWE</b>    | Signal write Enable (WE) de la bascule Bim, Si le signal est à 0 elle sera activée (fonctionnelle)                    |
| <b>BreWE</b>    | Signal write Enable (WE) de la bascule Bre, Si le signal est à 0 elle est activée <del>sur</del>                      |
| <b>MulRegWE</b> | Signal write Enable (WE) de la bascule mulReg qui recevra le résultat de la multiplication                            |
| <b>AddRegWE</b> | Signal write Enable de la bascule ADD_Reg qui recevra le résultat de l'addition.                                      |
| <b>CreWE</b>    | Signal write Enable de la bascule Cre   |
| <b>CinWE</b>    | Signal write Enable de la bascule Cin   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |



**Question 6 :** complétez diagramme de transition de l'unité de contrôle du circuit, en incluant la gestion des entrées/sorties.



www.scholarone.com



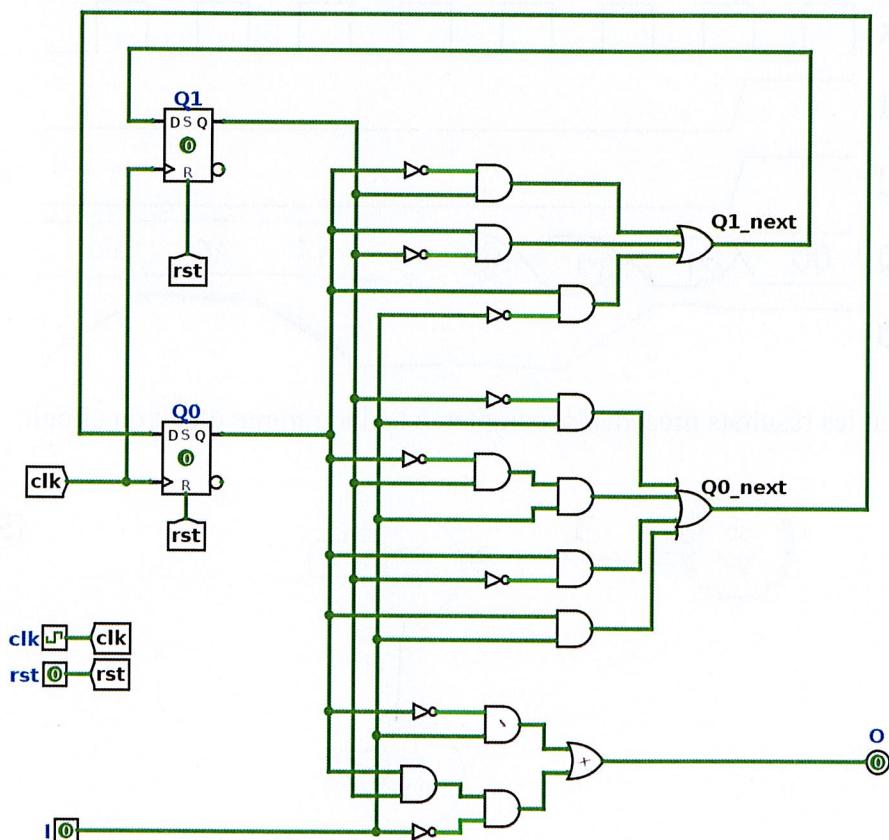
- Durée de l'épreuve : 2 heures
- Polycopié de cours autorisé, **autres documents interdits**, calculatrices non connectées autorisées

|            |  |
|------------|--|
| Exercice 1 |  |
| Exercice 2 |  |
| Total      |  |

**Attention, toutes les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.**

### Exercice 1 : machine à états (7 points)

On s'intéresse à un circuit séquentiel à une entrée  $I$  et une sortie  $O$ , et dont l'état interne est codé par un registre 2 bits  $Q[1:0]$ . Le registre dispose d'une commande  $rst$  (remise à zéro) qui est utilisée pour forcer l'état initial de la machine à 00.



**Q1 :** S'agit-il d'une machine de Mealy ou de Moore ? Justifiez votre réponse.

C'est une machine de Mealy car la sortie  $O$  dépend de l'entrée  $I$   
et des états ~~des~~ des registres  $Q_0$  et  $Q_1$  (basades  $Q_0$  et  $Q_1$ )



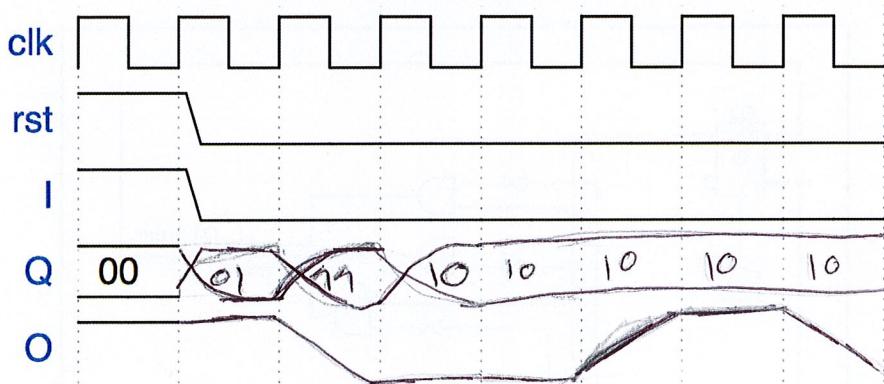
**Q3 :** Donner l'expression booléennes associée à  $Q_1^{t+1}$ .

$$Q_1^{t+1} = (\bar{Q}_0^t \cdot Q_1^t) + (Q_0^t \cdot \bar{Q}_1^t) + (Q_0^t \cdot I)$$

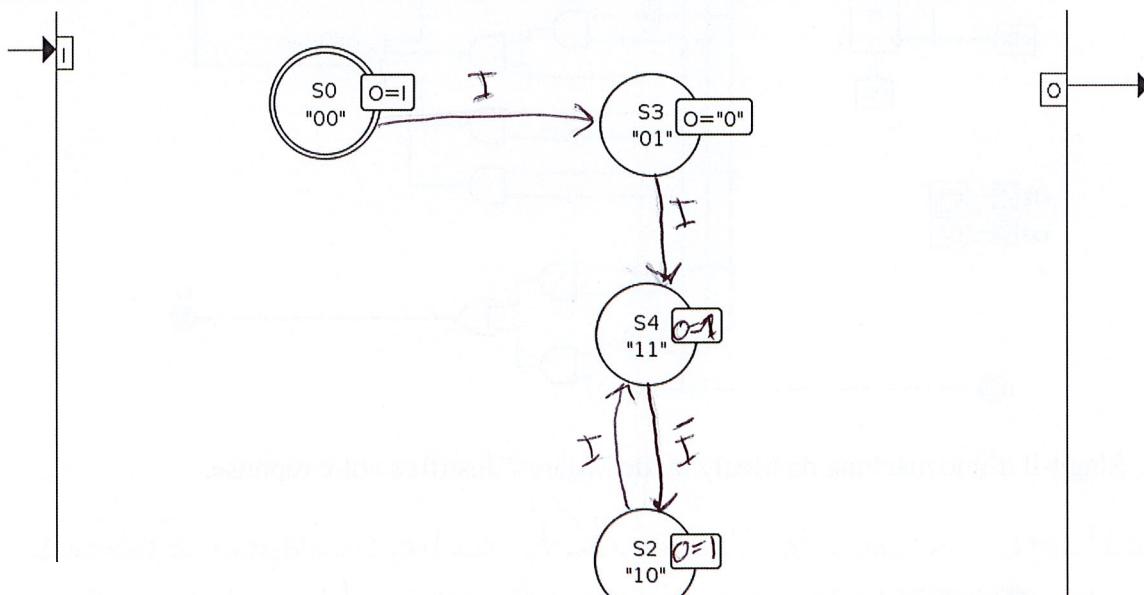
**Q4 :** Compléter la table de transition de la machine à état correspondant à ce circuit.

|   | $Q_1^t$ | $Q_0^t$ | $I^t$ | $Q_1^{t+1}$ | $Q_0^{t+1}$ | $O^t$ |
|---|---------|---------|-------|-------------|-------------|-------|
| 0 | 0       | 0       | 0     | 0           | 0           | 0     |
| 1 | 0       | 0       | 1     | 0           | 1           | 1     |
| 2 | 0       | 1       | 0     | 1           | 1           | 0     |
| 3 | 0       | 1       | 1     | 1           | 1           | 0     |
| 4 | 1       | 0       | 0     | 1           | 0           | 0     |
| 5 | 1       | 0       | 1     | 1           | 1           | 1     |
| 6 | 1       | 1       | 0     | 1           | 0           | 1     |
| 7 | 1       | 1       | 1     | 0           | 1           | 0     |

**Q5 :** Complétez le chronogramme avec les valeurs de Q et O.



**Q6 :** En utilisant les résultats précédents, complétez le diagramme d'état du circuit.



### Exercice 3 : décomposition UT/UC (13 points)

On souhaite réaliser un circuit permettant de calculer le produit de deux nombres complexes  $a$  et  $b$ , dont les parties réelles et imaginaires sont représentées par des entiers sur 16 bits. Le produit est calculé selon l'algorithme ci-dessous :

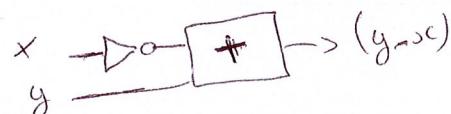
```
debut
    a_re := lire_entree();
    a_im := lire_entree();
    b_re := lire_entree();
    b_im := lire_entree();
    c_re := a_re.b_re-a_im.b_im;
    c_im := a_re.b_im+a_im.b_re;
    écrire_sortie(c_re);
    écrire_sortie(c_im);
fin;
```

Afin de réduire le nombre de portes logiques utilisées par le circuit, on fait le choix de limiter le nombre d'opérateurs arithmétique en n'utilisant qu'un seul additionneur et un seul multiplicateur, comme indiqué sur le circuit de la question 4.

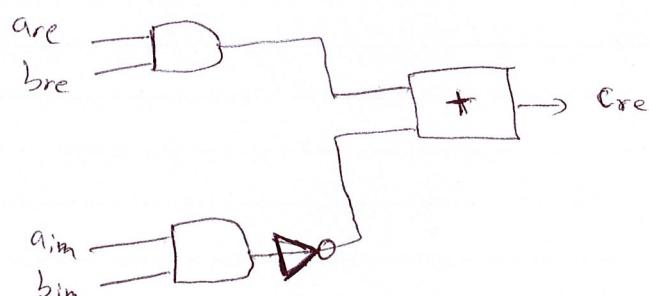
**Question 1 :** L'algorithme requiert des opérations de soustraction, or notre composant ne dispose que d'additionneurs. Expliquez (en vous aidant d'un schéma) comment il est possible de réaliser un circuit soustracteur à partir d'un additionneur et d'un ensemble de portes « inverseur ».

Il est possible de rajouter une porte logique pour la négation

en général:



sur notre exemple:



**Question 2 :** Le circuit de la question 4 permet de réaliser une addition et une multiplication en parallèle dans un même cycle. Utilisez cette propriété pour réécrire l'algorithme de multiplication de façon à minimiser le nombre d'étapes (cycles) de calcul. Vous utiliserez pour cela les conventions vues en cours et rappelées dans le tableau ci-après, et pourrez introduire des variables intermédiaires afin de faciliter la compréhension du code.

| Exécution    | Notation   | Signification  |
|--------------|--|--|
| Parallèle    | $\mathbf{X} = \mathbf{Y} + 1, \quad \mathbf{Y} = \mathbf{X} - 1;$  | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ et $\mathbf{Y}^{t+1} = \mathbf{X}^t - 1;$            |
| Séquentielle | $\mathbf{X} = \mathbf{Y} + 1 ;$<br>$\mathbf{Y} = \mathbf{X} - 1 ;$ | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ suivi de $\mathbf{Y}^{t+2} = \mathbf{X}^{t+1} - 1 ;$ |

```

début
    are := lire_entree();
    aim := lire_entree();
    bre := lire_entree();
    bim := lire_entree();

```

$$1 \quad \cancel{x^{t+1}} = y^t + 1, \quad \cancel{y^{t+1}} = x^t - 1$$

$$2 \quad \cancel{y^{t+1}} = x^t - 1;$$

$$3 \quad \cancel{x^{t+1}} = y^t + 1;$$

4

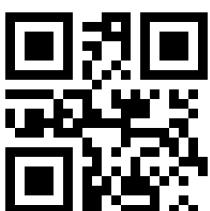
5

6

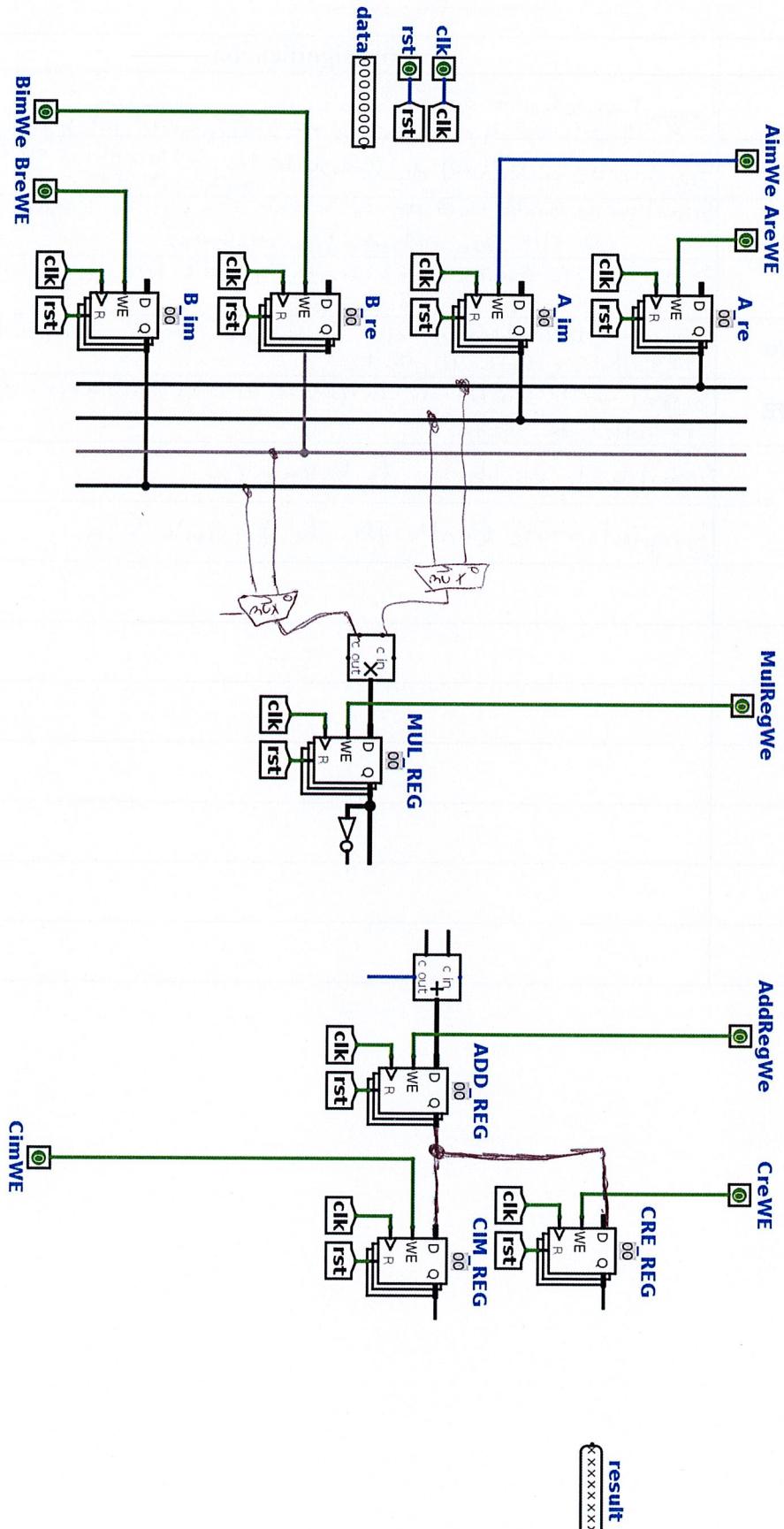
**fin;**

**Question 3 :** sachant que les entrées sorties se font sur le principe du protocole req/ack, combien de cycles faut-il au minimum pour réaliser l'ensemble des traitements (calculs et entrées/sorties) ? Veillez à bien justifier vos réponses

6 cycles



**Question 4 :** Utilisez les résultats des questions précédentes pour compléter le schéma de l'unité de traitement du circuit, en y ajoutant les interconnexions, multiplexeurs, etc. permettant de réaliser le traitement selon l'approche proposée aux questions 1 et 2.

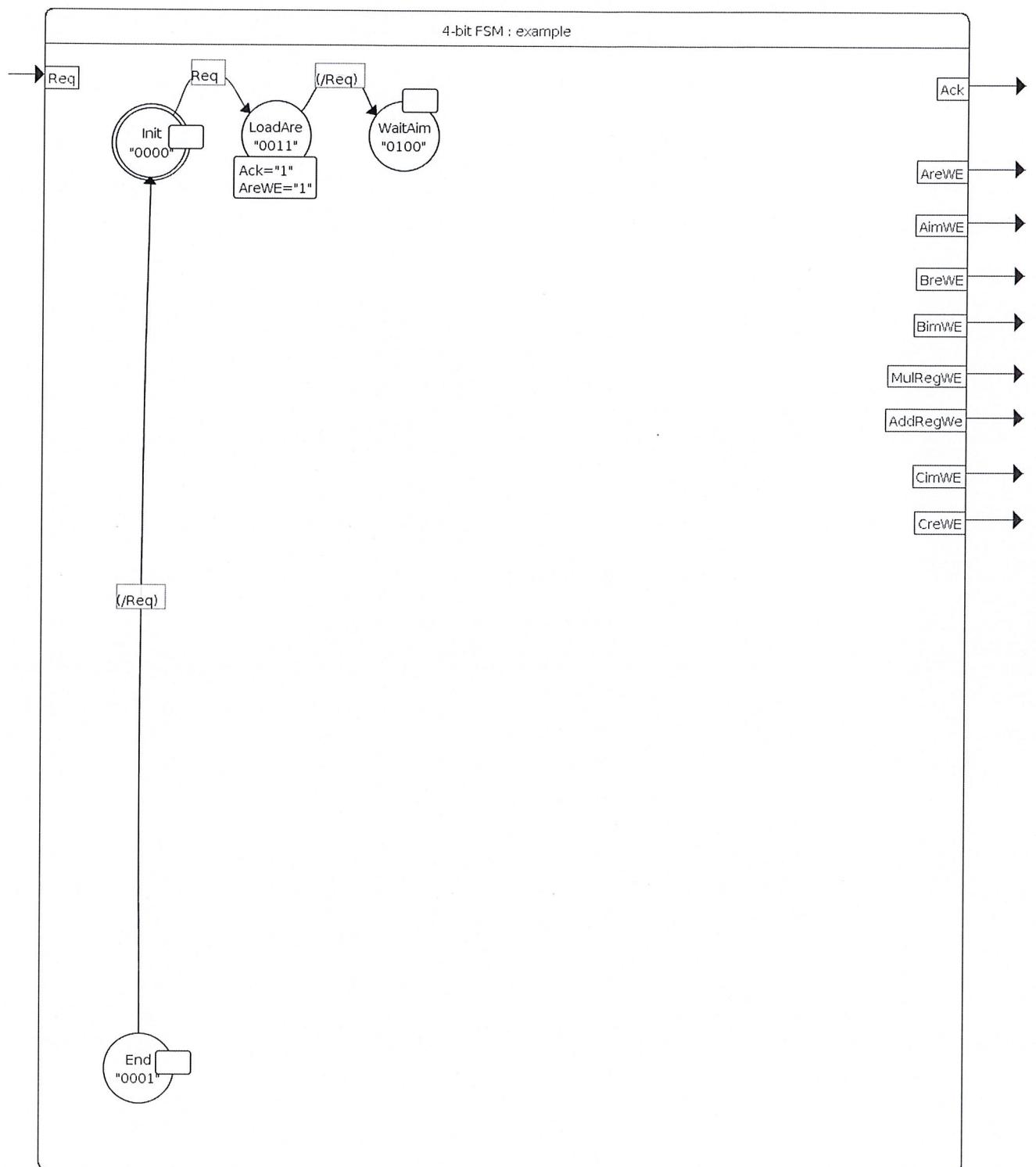


**Question 5 :** Complétez le tableau ci-dessous avec les signaux de contrôle supplémentaires, en explicitant à chaque fois leur rôle .

| Nom             | Rôle/signification  |
|-----------------|---|
| <b>AimWE</b>    | Signal write Enable de la bascule Aim, si elle est à 1 la bascule n'incremente pas (grâce au signal de top d'horloge) |
| <b>AreWE</b>    | Signal write Enable (WE) de la bascule Are, si le signal est à 0 elle sera activée                                    |
| <b>BimWE</b>    | Signal write Enable (WE) de la bascule Bim, Si le signal est à 0 elle sera activée (fonctionnelle)                    |
| <b>BreWE</b>    | Signal write Enable (WE) de la bascule Bre, Si le signal est à 0 elle est activée <del>sur</del>                      |
| <b>MulRegWE</b> | Signal write Enable (WE) de la bascule mulReg qui recevra le résultat de la multiplication                            |
| <b>AddRegWE</b> | Signal write Enable de la bascule ADD_Reg qui recevra le résultat de l'addition                                       |
| <b>CreWE</b>    | Signal write Enable de la bascule Cre   |
| <b>CinWE</b>    | Signal write Enable de la bascule Cin   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |



**Question 6 :** complétez diagramme de transition de l'unité de contrôle du circuit, en incluant la gestion des entrées/sorties.



http://www.jingyan.com/jingyan/2013/09/20130912101355555.htm



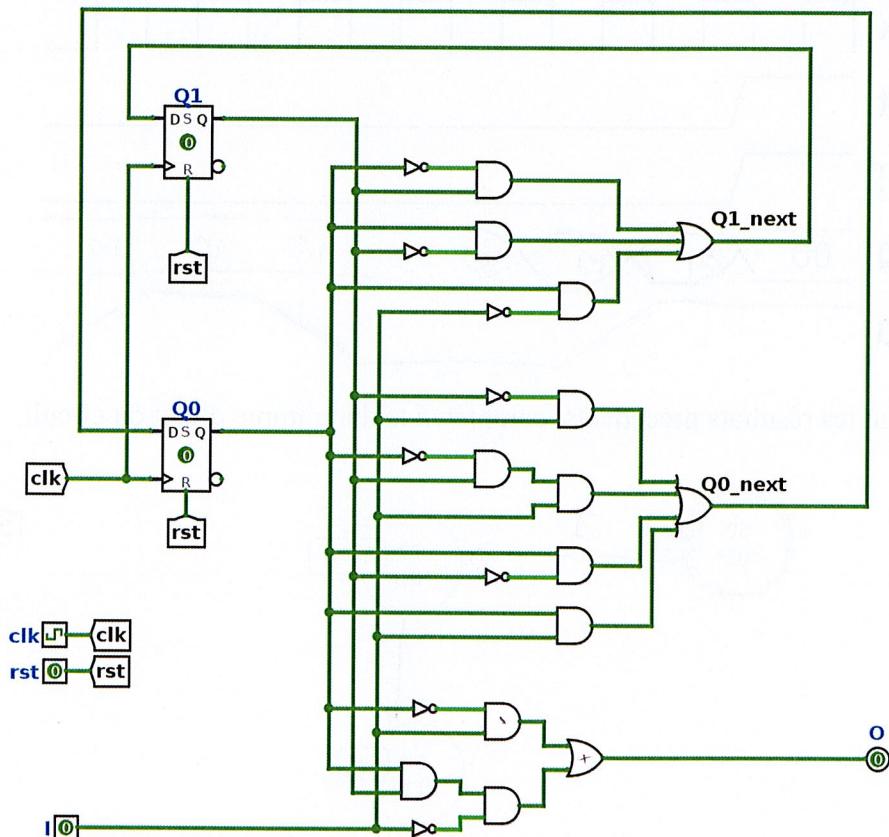
- Durée de l'épreuve : 2 heures
- Polycopié de cours autorisé, **autres documents interdits**, calculatrices non connectées autorisées

|            |  |
|------------|--|
| Exercice 1 |  |
| Exercice 2 |  |
| Total      |  |

**Attention, toutes les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.**

### Exercice 1 : machine à états (7 points)

On s'intéresse à un circuit séquentiel à une entrée I et une sortie O, et dont l'état interne est codé par un registre 2 bits Q[1:0]. Le registre dispose d'une commande rst (remise à zéro) qui est utilisée pour forcer l'état initial de la machine à 00.



**Q1 :** S'agit-il d'une machine de Mealy ou de Moore ? Justifiez votre réponse.

C'est une machine de Mealy car la sortie O dépend l'entrée I  
es ~~évolue~~ des registres Q<sub>0</sub> et Q<sub>1</sub> (basades Q<sub>0</sub> et Q<sub>1</sub>)



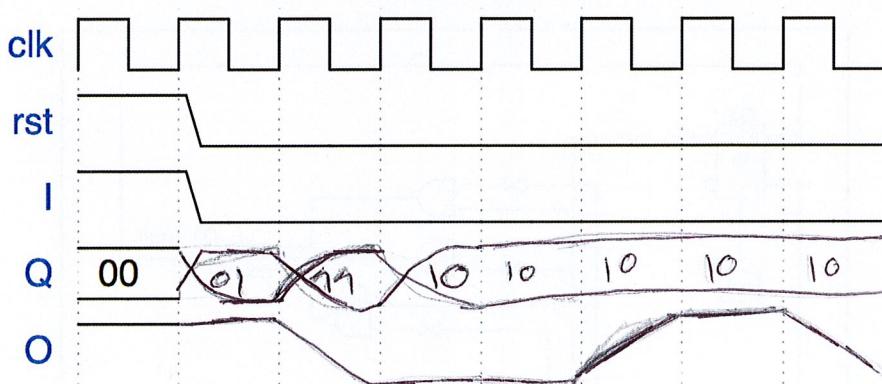
**Q3 :** Donner l'expression booléennes associée à  $Q_1^{t+1}$ .

$$Q_1^{t+1} = (\bar{Q}_0^t \cdot Q_1^t) + (Q_0^t \cdot \bar{Q}_1^t) + (Q_0^t \cdot I)$$

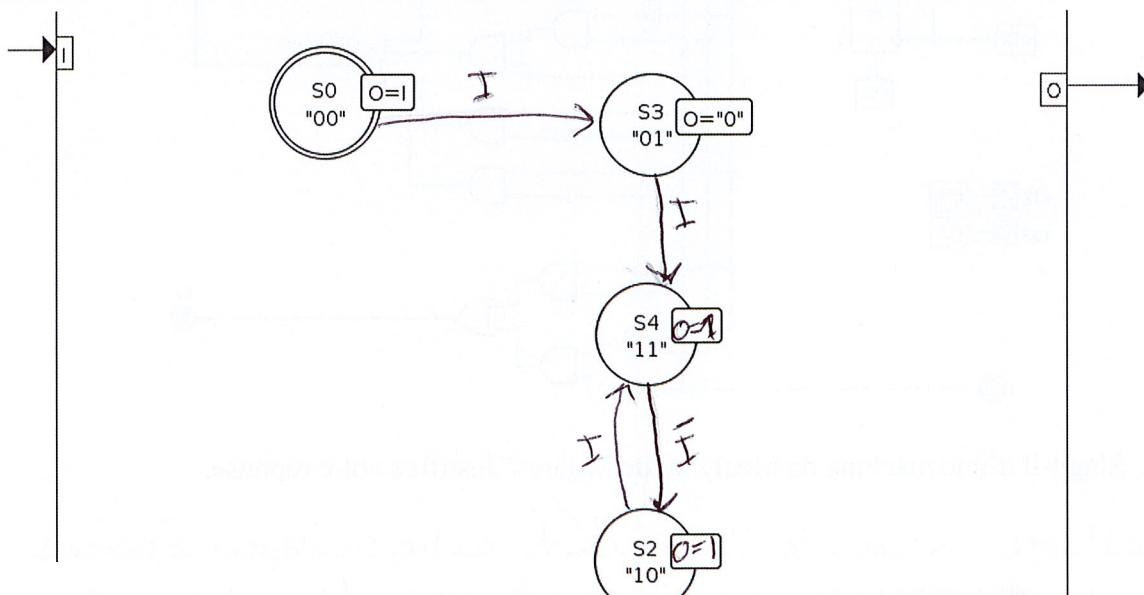
**Q4 :** Compléter la table de transition de la machine à état correspondant à ce circuit.

|   | $Q_1^t$ | $Q_0^t$ | $I^t$ | $Q_1^{t+1}$ | $Q_0^{t+1}$ | $O^t$ |
|---|---------|---------|-------|-------------|-------------|-------|
| 0 | 0       | 0       | 0     | 0           | 0           | 0     |
| 1 | 0       | 0       | 1     | 0           | 1           | 1     |
| 2 | 0       | 1       | 0     | 1           | 1           | 0     |
| 3 | 0       | 1       | 1     | 1           | 1           | 0     |
| 4 | 1       | 0       | 0     | 1           | 0           | 0     |
| 5 | 1       | 0       | 1     | 1           | 1           | 1     |
| 6 | 1       | 1       | 0     | 1           | 0           | 1     |
| 7 | 1       | 1       | 1     | 0           | 1           | 0     |

**Q5 :** Complétez le chronogramme avec les valeurs de Q et O.



**Q6 :** En utilisant les résultats précédents, complétez le diagramme d'état du circuit.



### Exercice 3 : décomposition UT/UC (13 points)

On souhaite réaliser un circuit permettant de calculer le produit de deux nombres complexes  $a$  et  $b$ , dont les parties réelles et imaginaires sont représentées par des entiers sur 16 bits. Le produit est calculé selon l'algorithme ci-dessous :

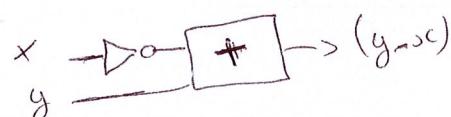
```
debut
    a_re := lire_entree();
    a_im := lire_entree();
    b_re := lire_entree();
    b_im := lire_entree();
    c_re := a_re.b_re-a_im.b_im;
    c_im := a_re.b_im+a_im.b_re;
    écrire_sortie(c_re);
    écrire_sortie(c_im);
fin;
```

Afin de réduire le nombre de portes logiques utilisées par le circuit, on fait le choix de limiter le nombre d'opérateurs arithmétique en n'utilisant qu'un seul additionneur et un seul multiplicateur, comme indiqué sur le circuit de la question 4.

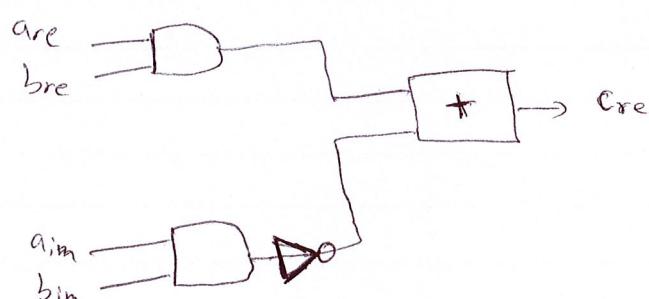
**Question 1 :** L'algorithme requiert des opérations de soustraction, or notre composant ne dispose que d'additionneurs. Expliquez (en vous aidant d'un schéma) comment il est possible de réaliser un circuit soustracteur à partir d'un additionneur et d'un ensemble de portes « inverseur ».

Il est possible de rajouter une porte logique pour la négation

en général:



sur notre exemple:



**Question 2 :** Le circuit de la question 4 permet de réaliser une addition et une multiplication en parallèle dans un même cycle. Utilisez cette propriété pour réécrire l'algorithme de multiplication de façon à minimiser le nombre d'étapes (cycles) de calcul. Vous utiliserez pour cela les conventions vues en cours et rappelées dans le tableau ci-après, et pourrez introduire des variables intermédiaires afin de faciliter la compréhension du code.

| Exécution    | Notation   | Signification  |
|--------------|--|--|
| Parallèle    | $\mathbf{X} = \mathbf{Y} + 1$ , $\mathbf{Y} = \mathbf{X} - 1$ ;    | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ et $\mathbf{Y}^{t+1} = \mathbf{X}^t - 1$ ;           |
| Séquentielle | $\mathbf{X} = \mathbf{Y} + 1$ ;<br>$\mathbf{Y} = \mathbf{X} - 1$ ; | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ suivi de $\mathbf{Y}^{t+2} = \mathbf{X}^{t+1} - 1$ ; |

```

début
    are := lire_entree();
    aim := lire_entree();
    bre := lire_entree();
    bim := lire_entree();

```

$$1 \quad \cancel{x^{t+1}} = y^t + 1, \cancel{y^{t+1}} = x^t - 1$$

$$2 \quad \cancel{y^{t+1}} = x^t - 1;$$

$$3 \quad \cancel{x^{t+1}} = y^t + 1;$$

4

5

6

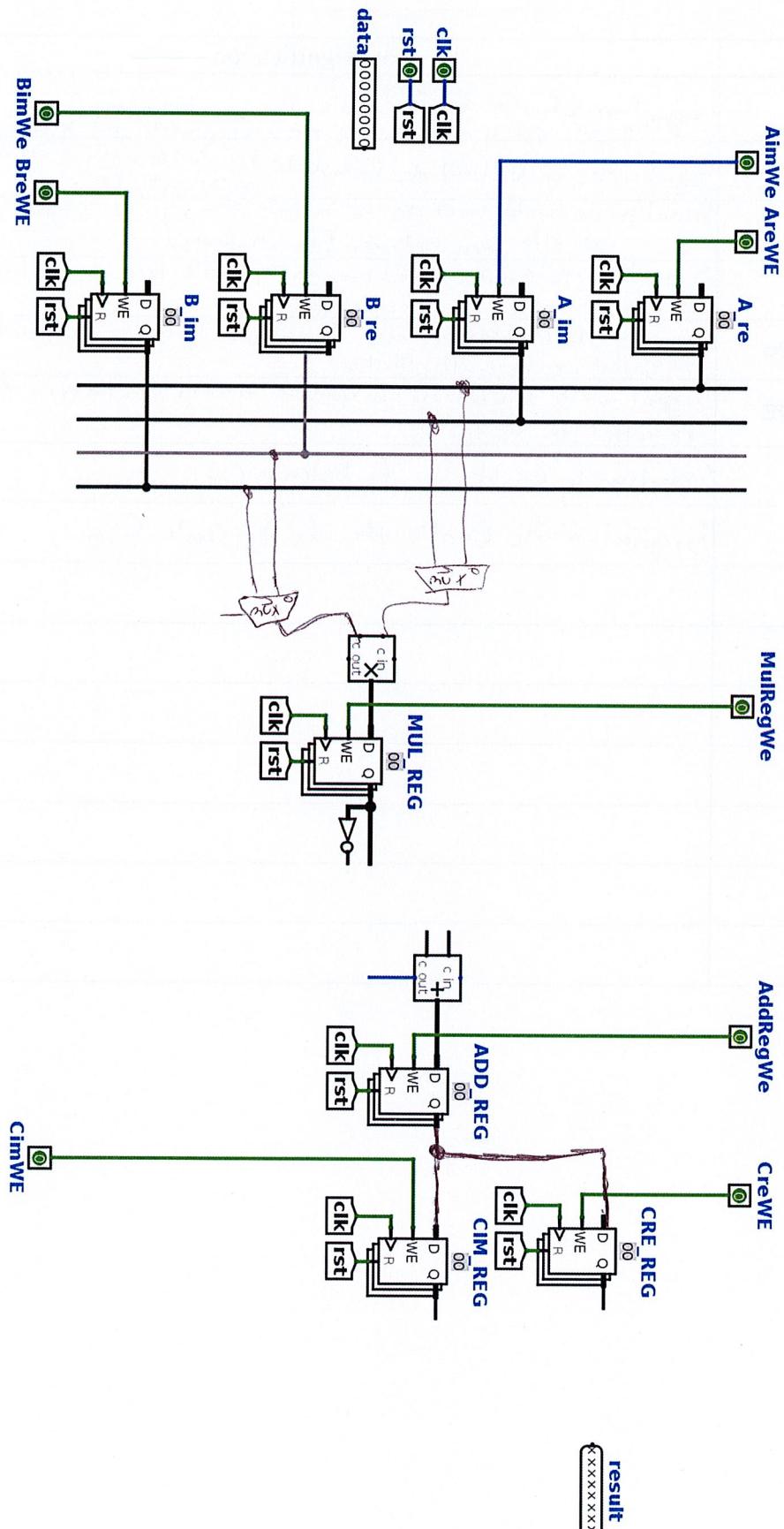
**fin;**

**Question 3 :** sachant que les entrées sorties se font sur le principe du protocole req/ack, combien de cycles faut-il au minimum pour réaliser l'ensemble des traitements (calculs et entrées/sorties) ? Veillez à bien justifier vos réponses

6 cycles



**Question 4 :** Utilisez les résultats des questions précédentes pour compléter le schéma de l'unité de traitement du circuit, en y ajoutant les interconnexions, multiplexeurs, etc. permettant de réaliser le traitement selon l'approche proposée aux questions 1 et 2.

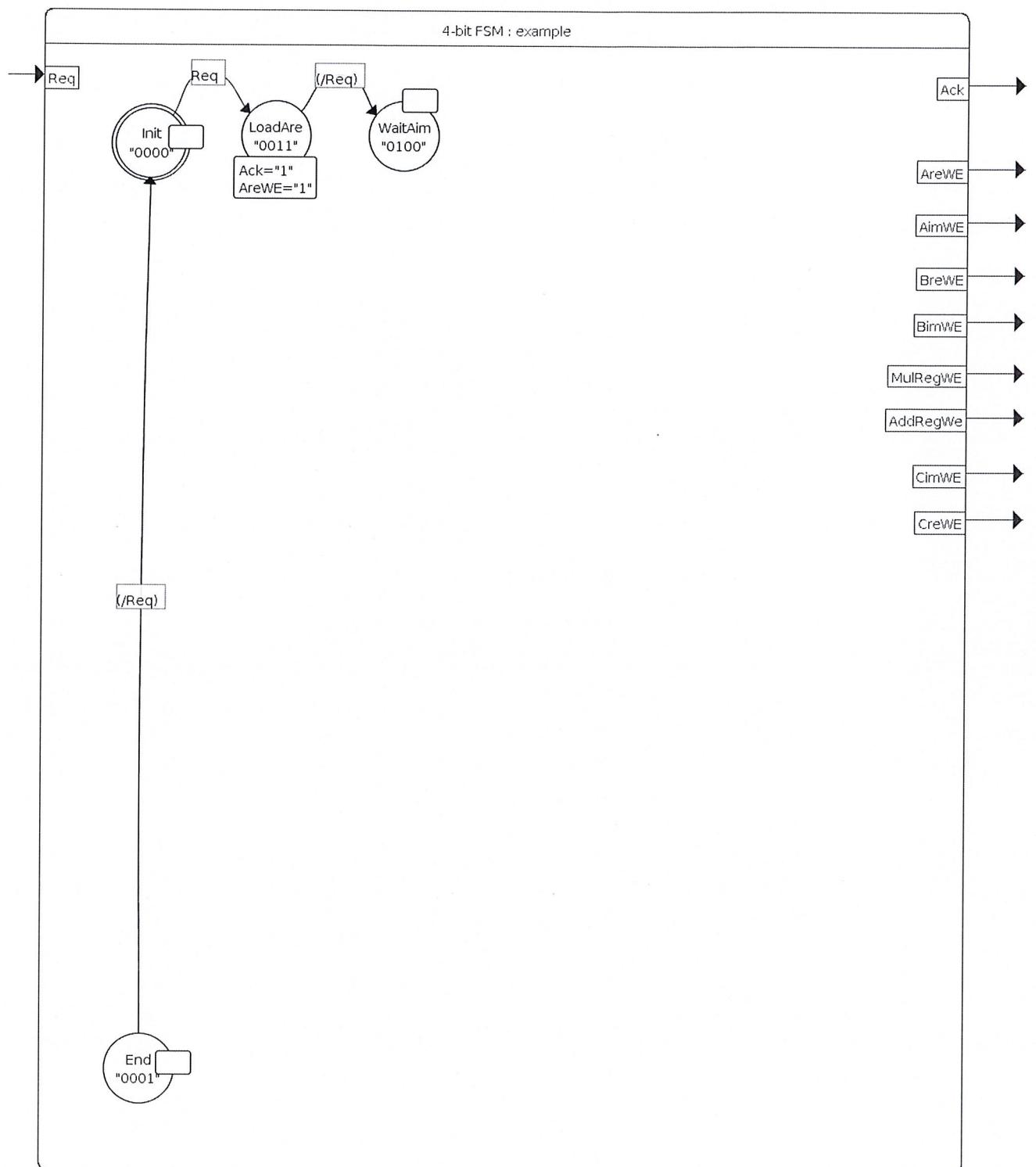


**Question 5 :** Complétez le tableau ci-dessous avec les signaux de contrôle supplémentaires, en explicitant à chaque fois leur rôle .

| Nom             | Rôle/signification  |
|-----------------|---|
| <b>AimWE</b>    | Signal write Enable de la bascule Aim, si elle est à 1 la bascule n'incremente pas (grâce au signal de top d'horloge) |
| <b>AreWE</b>    | Signal write Enable (WE) de la bascule Are, si le signal est à 0 elle sera activée                                    |
| <b>BimWE</b>    | Signal write Enable (WE) de la bascule Bim, Si le signal est à 0 elle sera activée (fonctionnelle)                    |
| <b>BreWE</b>    | Signal write Enable (WE) de la bascule Bre, Si le signal est à 0 elle est activée <del>sur</del>                      |
| <b>MulRegWE</b> | Signal write Enable (WE) de la bascule mulReg qui recevra le résultat de la multiplication                            |
| <b>AddRegWE</b> | Signal write Enable de la bascule ADD_Reg qui recevra le résultat de l'addition.                                      |
| <b>CreWE</b>    | Signal write Enable de la bascule Cre   |
| <b>CinWE</b>    | Signal write Enable de la bascule Cin   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |



**Question 6 :** complétez diagramme de transition de l'unité de contrôle du circuit, en incluant la gestion des entrées/sorties.





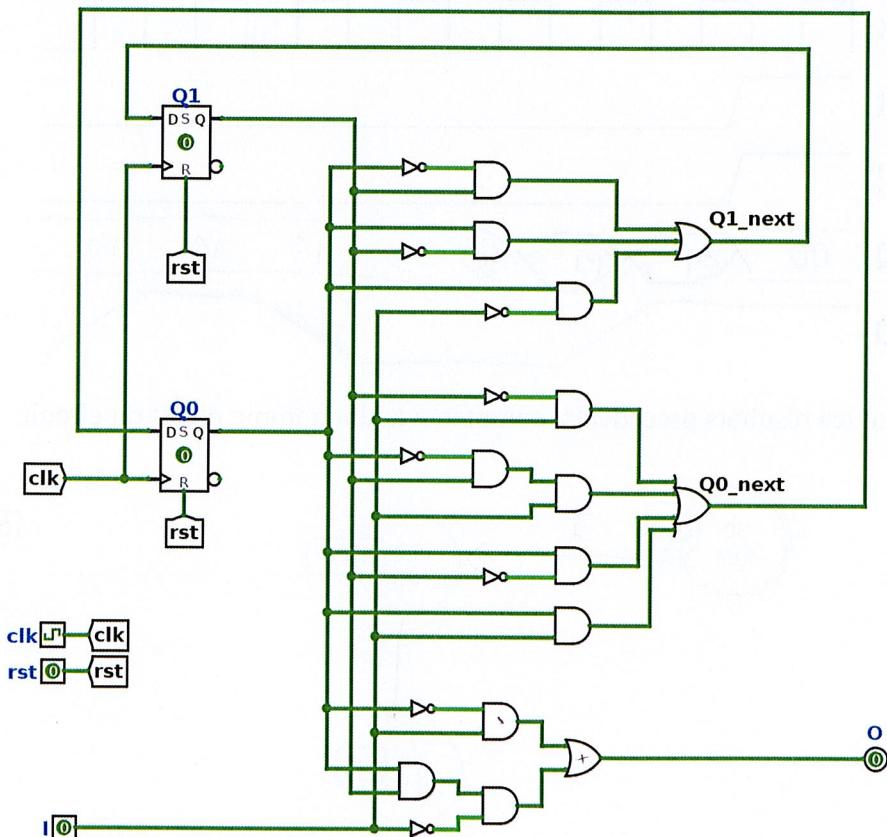
- Durée de l'épreuve : 2 heures
- Polycopié de cours autorisé, **autres documents interdits**, calculatrices non connectées autorisées

|            |  |
|------------|--|
| Exercice 1 |  |
| Exercice 2 |  |
| Total      |  |

**Attention, toutes les réponses doivent être portées sur ce document, sous peine d'être ignorées par le correcteur.**

### Exercice 1 : machine à états (7 points)

On s'intéresse à un circuit séquentiel à une entrée  $I$  et une sortie  $O$ , et dont l'état interne est codé par un registre 2 bits  $Q[1:0]$ . Le registre dispose d'une commande  $rst$  (remise à zéro) qui est utilisée pour forcer l'état initial de la machine à 00.



**Q1 :** S'agit-il d'une machine de Mealy ou de Moore ? Justifiez votre réponse.

C'est une machine de Mealy car la sortie  $O$  dépend de l'entrée  $I$   
et ~~des états~~ des registres  $Q_0$  et  $Q_1$  (basades  $Q_0$  et  $Q_1$ )



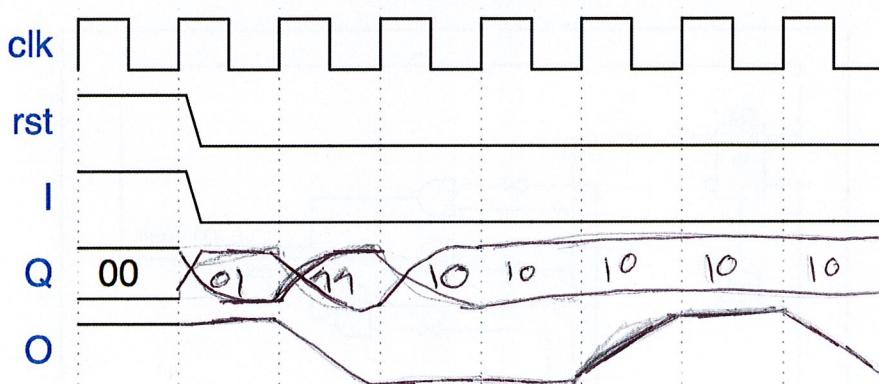
Q3 : Donner l'expression booléennes associée à  $Q_1^{t+1}$ .

$$Q_1^{t+1} = (\bar{Q}_0^t \cdot Q_1^t) + (Q_0^t \cdot \bar{Q}_1^t) + (Q_0^t \cdot I)$$

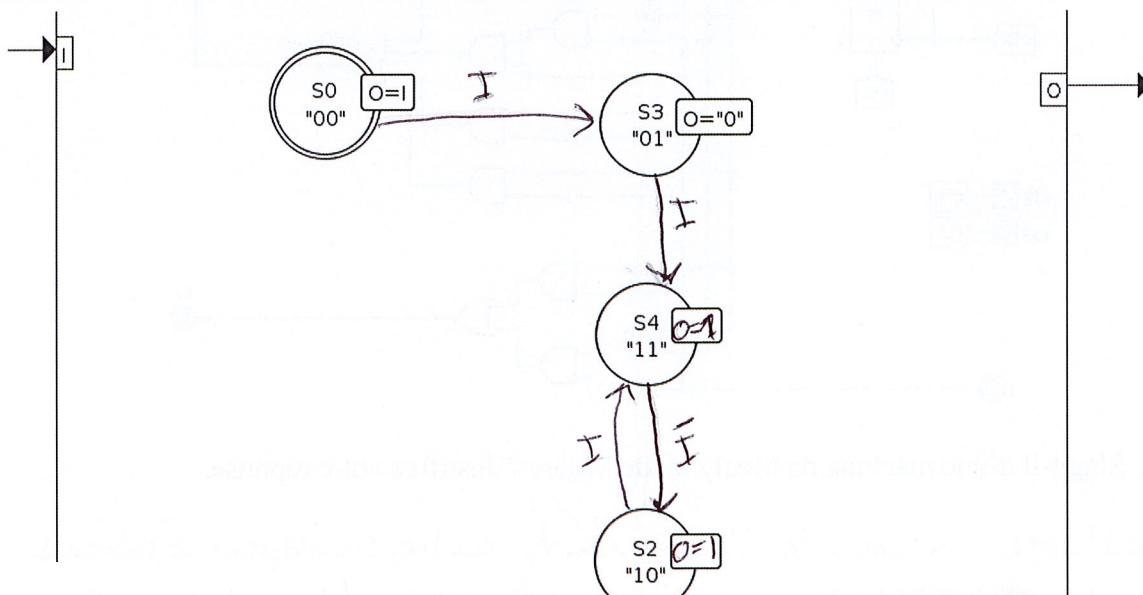
Q4 : Compléter la table de transition de la machine à état correspondant à ce circuit.

|   | $Q_1^t$ | $Q_0^t$ | $I^t$ | $Q_1^{t+1}$ | $Q_0^{t+1}$ | $O^t$ |
|---|---------|---------|-------|-------------|-------------|-------|
| 0 | 0       | 0       | 0     | 0           | 0           | 0     |
| 1 | 0       | 0       | 1     | 0           | 1           | 1     |
| 2 | 0       | 1       | 0     | 1           | 1           | 0     |
| 3 | 0       | 1       | 1     | 1           | 1           | 0     |
| 4 | 1       | 0       | 0     | 1           | 0           | 0     |
| 5 | 1       | 0       | 1     | 1           | 1           | 1     |
| 6 | 1       | 1       | 0     | 1           | 0           | 1     |
| 7 | 1       | 1       | 1     | 0           | 1           | 0     |

Q5 : Complétez le chronogramme avec les valeurs de Q et O.



Q6 : En utilisant les résultats précédents, complétez le diagramme d'état du circuit.



### Exercice 3 : décomposition UT/UC (13 points)

On souhaite réaliser un circuit permettant de calculer le produit de deux nombres complexes  $a$  et  $b$ , dont les parties réelles et imaginaires sont représentées par des entiers sur 16 bits. Le produit est calculé selon l'algorithme ci-dessous :

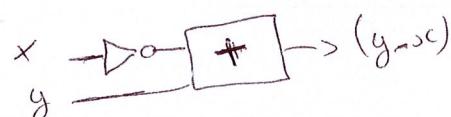
```
debut
    a_re := lire_entree();
    a_im := lire_entree();
    b_re := lire_entree();
    b_im := lire_entree();
    c_re := a_re.b_re-a_im.b_im;
    c_im := a_re.b_im+a_im.b_re;
    écrire_sortie(c_re);
    écrire_sortie(c_im);
fin;
```

Afin de réduire le nombre de portes logiques utilisées par le circuit, on fait le choix de limiter le nombre d'opérateurs arithmétique en n'utilisant qu'un seul additionneur et un seul multiplicateur, comme indiqué sur le circuit de la question 4.

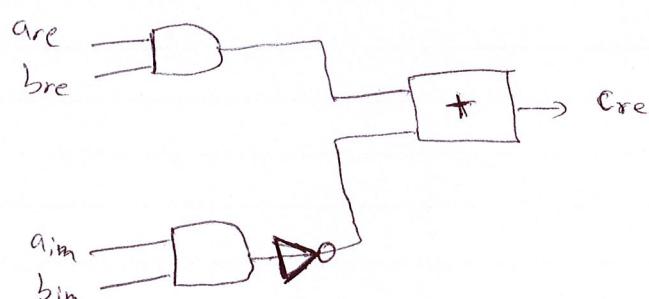
**Question 1 :** L'algorithme requiert des opérations de soustraction, or notre composant ne dispose que d'additionneurs. Expliquez (en vous aidant d'un schéma) comment il est possible de réaliser un circuit soustracteur à partir d'un additionneur et d'un ensemble de portes « inverseur ».

Il est possible de rajouter une porte logique pour la négation

en général:



sur notre exemple:



**Question 2 :** Le circuit de la question 4 permet de réaliser une addition et une multiplication en parallèle dans un même cycle. Utilisez cette propriété pour réécrire l'algorithme de multiplication de façon à minimiser le nombre d'étapes (cycles) de calcul. Vous utiliserez pour cela les conventions vues en cours et rappelées dans le tableau ci-après, et pourrez introduire des variables intermédiaires afin de faciliter la compréhension du code.

| Exécution    | Notation   | Signification  |
|--------------|--|--|
| Parallèle    | $\mathbf{X} = \mathbf{Y} + 1$ , $\mathbf{Y} = \mathbf{X} - 1$ ;    | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ et $\mathbf{Y}^{t+1} = \mathbf{X}^t - 1$ ;           |
| Séquentielle | $\mathbf{X} = \mathbf{Y} + 1$ ;<br>$\mathbf{Y} = \mathbf{X} - 1$ ; | $\mathbf{X}^{t+1} = \mathbf{Y}^t + 1$ suivi de $\mathbf{Y}^{t+2} = \mathbf{X}^{t+1} - 1$ ; |

```

début
    are := lire_entree();
    aim := lire_entree();
    bre := lire_entree();
    bim := lire_entree();

```

$$\begin{array}{l}
 1 \quad \cancel{x^{t+1}} = y^t + 1, \cancel{y^{t+1}} = x^t - 1 \\
 2 \quad \cancel{y^{t+1}} = x^t - 1; \\
 3 \quad \cancel{x^{t+1}} = y^t + 1; \\
 4 \quad \underline{\hspace{1cm}} \\
 5 \quad \underline{\hspace{1cm}} \\
 6 \quad \underline{\hspace{1cm}}
 \end{array}$$

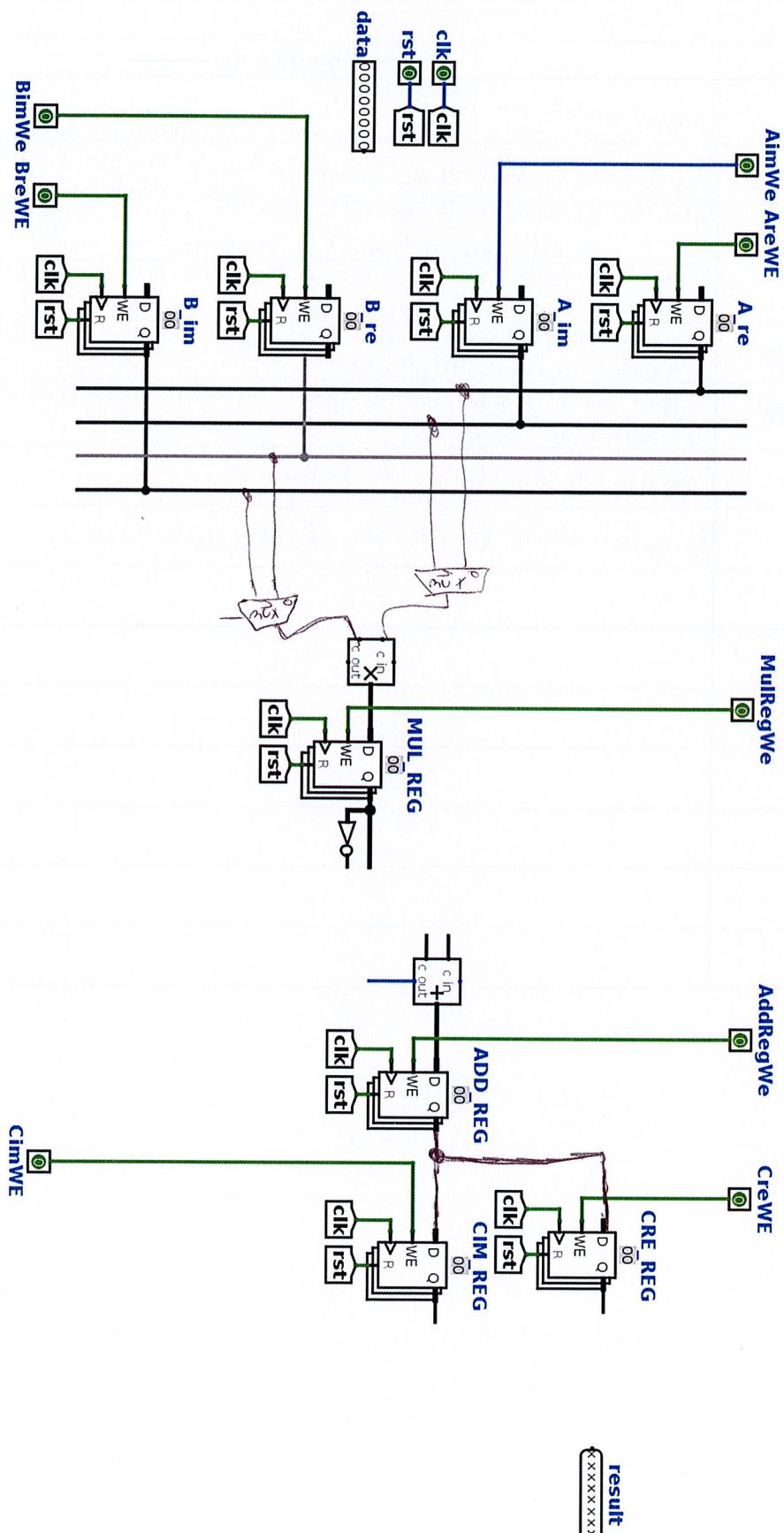
```
fin;
```

**Question 3 :** sachant que les entrées sorties se font sur le principe du protocole req/ack, combien de cycles faut-il au minimum pour réaliser l'ensemble des traitements (calculs et entrées/sorties) ? Veillez à bien justifier vos réponses

6 cycles



**Question 4 :** Utilisez les résultats des questions précédentes pour compléter le schéma de l'unité de traitement du circuit, en y ajoutant les interconnexions, multiplexeurs, etc. permettant de réaliser le traitement selon l'approche proposée aux questions 1 et 2.

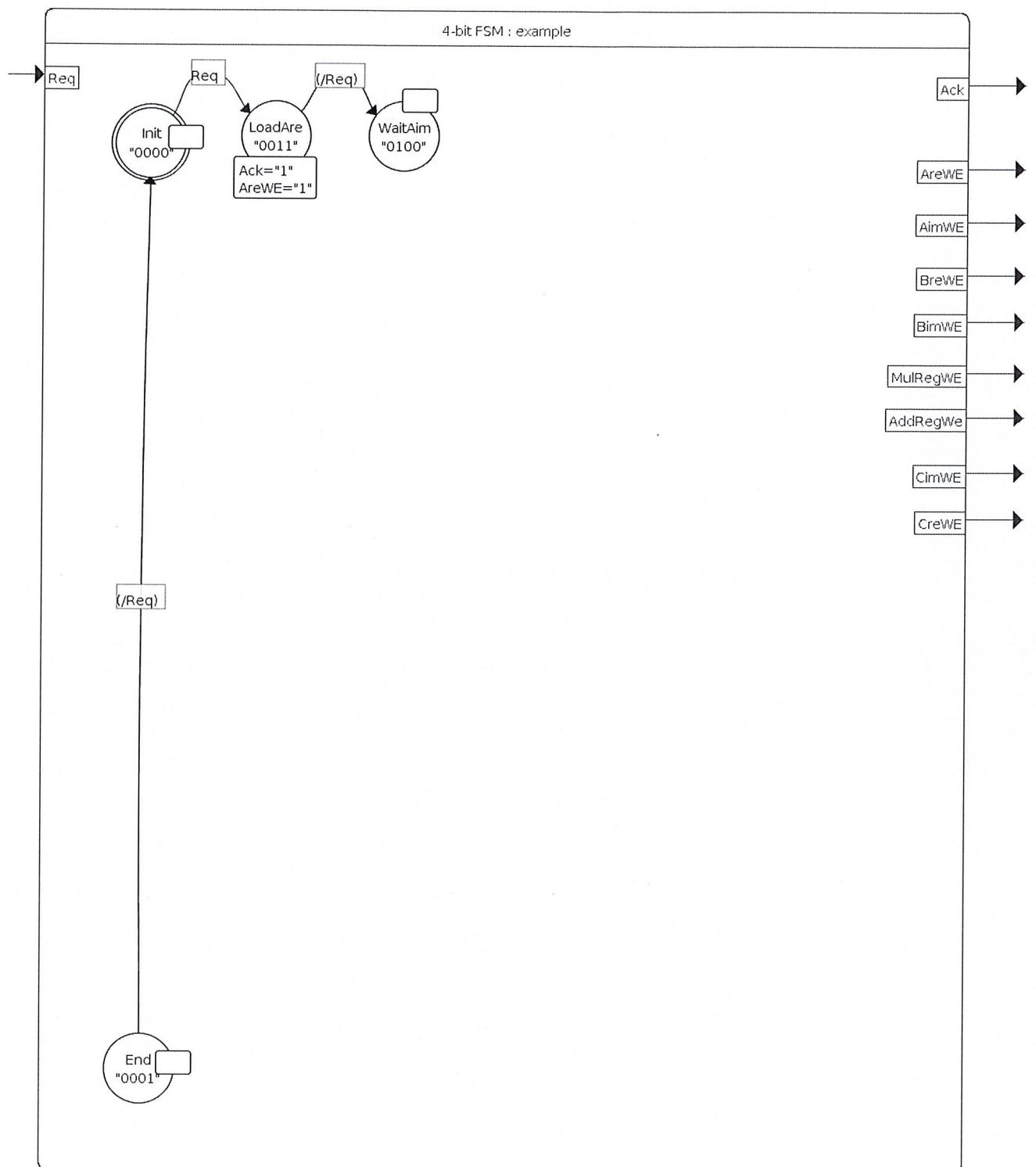


**Question 5 :** Complétez le tableau ci-dessous avec les signaux de contrôle supplémentaires, en explicitant à chaque fois leur rôle .

| Nom             | Rôle/signification  |
|-----------------|---|
| <b>AimWE</b>    | Signal write Enable de la bascule Aim, si elle est à 1 la bascule n'incremente pas (grâce au signal de top d'horloge) |
| <b>AreWE</b>    | Signal write Enable (WE) de la bascule Are, si le signal est à 0 elle sera activée                                    |
| <b>BimWE</b>    | Signal write Enable (WE) de la bascule Bim, Si le signal est à 0 elle sera activée (fonctionnelle)                    |
| <b>BreWE</b>    | Signal write Enable (WE) de la bascule Bre, Si le signal est à 0 elle est activée <del>sur</del>                      |
| <b>MulRegWE</b> | Signal write Enable (WE) de la bascule mulReg qui recevra le résultat de la multiplication                            |
| <b>AddRegWE</b> | Signal write Enable de la bascule ADD_Reg qui recevra le résultat de l'addition                                       |
| <b>CreWE</b>    | Signal write Enable de la bascule Cre   |
| <b>CinWE</b>    | Signal write Enable de la bascule Cin   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |
|                 |   |



**Question 6 :** complétez diagramme de transition de l'unité de contrôle du circuit, en incluant la gestion des entrées/sorties.



http://www.jingyan.com/jingyan/zhongguo/zhongguo\_zhongguo/zhongguo\_zhongguo\_001.htm

