

# Composition d'automatique et d'informatique industrielle

durée 6 heures

## NOTATIONS - VALEURS NUMERIQUES

- $N$  : nombre de spires par branche du bobinage de polarisation.
- $n_1$  : nombre de spires par branche du bobinage 1.
- $n_2$  : nombre de spires par branche du bobinage 2.
- $\mu_0$  : perméabilité magnétique du vide ( $\approx$  perméabilité magnétique de l'air).
- $\mu_r$  : perméabilité magnétique relative du matériau du cylindre.
- $\mathcal{R}_A, \mathcal{R}_B, \mathcal{R}_C$  : réluctance des entrefers repérés par les points  $A, B, C$ .
- $r$  : réluctance du cylindre.
- $\mathcal{R}_0$  : réluctance de l'entrefer lorsqu'il est égal à  $e$ .
- $\varphi_A, \varphi_B, \varphi_C, \varphi_D, \varphi_E, \varphi_F, \varphi_G, \varphi_H$  : flux magnétiques partiels qui traversent respectivement les entrefers  $A, B, C, D, E, F, G, H$ .
- $\phi_A, \phi_B, \phi_C, \phi_D, \phi_E, \phi_F, \phi_G, \phi_H$  : flux magnétiques totaux qui traversent respectivement les entrefers  $A, B, C, D, E, F, G, H$ .
- $D$  : diamètre du cylindre.
- $S$  : section d'une branche à travers laquelle le flux magnétique passe.
- $S_1$  : section à travers laquelle le flux magnétique passe dans le cylindre.
- $e$  : entrefer (distance séparant le cylindre d'une branche) lorsque le cylindre ne vibre pas.
- $e_A, e_B, e_C, e_D, e_E, e_F, e_G, e_H$  : entrefer au niveau des branches notées respectivement  $A, B, C, D, E, F, G, H$ .
- $v_1(t)$  : amplitude de la vibration du cylindre selon l'axe  $AE$  (variation de la position du cylindre par rapport à sa position de repos).
- $v_2(t)$  : amplitude de la vibration du cylindre selon l'axe  $HD$  (variation de la position du cylindre par rapport à sa position de repos).
- $\dot{v}_1(t)$  : vitesse de vibration du cylindre selon l'axe  $AE$ .
- $\dot{v}_2(t)$  : vitesse de vibration du cylindre selon l'axe  $HD$ .
- $I$  : courant qui circule dans la bobine de polarisation.
- $\omega_0$  : pulsation naturelle du cylindre ( $2\pi \cdot 7800 \text{ rad.s}^{-1}$ ).
- $\Omega$  : vitesse angulaire de rotation du gyromètre.
- $k_c$  : coefficient de Coriolis ( $5,581162 \cdot 10^{-3} \text{ kg}$ ).
- $M$  : masse équivalente du cylindre ( $7,513045 \cdot 10^{-3} \text{ kg}$ ).
- $Q$  : coefficient de qualité du cylindre (4200).
- $f(X_0)$  : fonction caractérisant la déformation axiale du cylindre ( $X_0 = \frac{x_0}{H}$ ) ( $f(X_0) = 3,7052$ ).
- $G_{exc}$  : gain du bobinage d'excitation ( $G_{exc1} = G_{exc2} = 0,08$ ).
- $G_{det}$  : gain du bobinage de détection ( $G_{det1} = G_{det2} = 0,08$ ).
- $p$  : variable de Laplace.
- $V_1(p)$  : transformée de Laplace du signal  $v_1(t)$ .
- Nous utiliserons des lettres majuscules pour exprimer la transformée de Laplace d'un signal.
- $(\dots)^T$  : en notation vectorielle, "T" désigne le vecteur transposé. Il en est de même pour une matrice.

# Introduction

Ce devoir traite de la mise en oeuvre d'un capteur : un gyromètre. Il est utilisé pour mesurer une vitesse de rotation autour de son axe. Les gyromètres sont composés généralement d'un élément sensible accompagné d'une électronique de commande. L'objectif étant d'obtenir un signal de mesure proportionnel à la vitesse de rotation ainsi qu'une grande précision de mesure. Afin de diminuer les non linéarités introduites par exemple par un défaut de fabrication "mécanique", nous utilisons une structure bouclée.

Le devoir est composé de trois parties indépendantes :

- Modélisation d'un gyromètre mécanique vibrant ;
- Commande du gyromètre ;
- Implantation du dispositif de commande dans un composant programmable.

## PARTIE 1 Modélisation d'un gyromètre mécanique vibrant

La modélisation du gyromètre mécanique vibrant est décomposée en cinq sous-parties.



FIG. 1 – Vue générale du gyromètre

### 1.1 Bobinage de polarisation

Le bobinage de polarisation est parcouru par un courant continu noté  $I$ , permettant de créer un champ magnétique continu. Il possède  $N$  spires par branche. Le bobinage est montré sur la figure (6).

Le circuit magnétique est constitué de huit branches (pour le stator), par un cylindre (le "rotor") et par un entrefer. Le flux magnétique ainsi créé circule d'une branche à l'autre en passant par le cylindre.

Nous ferons les hypothèses suivantes :

- le cylindre est parfait,
- pas de flux de fuite.

Pour des raisons de symétrie, le circuit magnétique total peut être décomposé en huit circuits magnétiques élémentaires comme indiqué sur la figure (7). Nous supposons la perméabilité magnétique des branches du stator infinie.



FIG. 2 – Vue détaillée du gyromètre : à gauche le boîtier, au centre le cylindre vibrant et à droite les différents bobinages

**1. 1. 1.** Donner la définition du théorème d'Ampère.

**1. 1. 2.** Pour un bobinage isolé, donner la relation qui lie :  $U_{mm}$  (force magnétomotrice),  $\mathcal{R}$  (réductance du tube de flux fermé, associé au bobinage) et  $\varphi$  (flux magnétique).

**1. 1. 3.** A partir du théorème d'Ampère, calculer les réductances  $\mathcal{R}_A$ ,  $\mathcal{R}_B$  respectivement des entrefers repérés par les points A et B et la réductance  $r$  du cylindre. Nous considérerons que  $S_1 = S$ .

Pour l'étude du circuit magnétique de la figure (7), il est également possible d'utiliser les analogies électriques données sur le document réponse (1).

**1. 1. 4.** Compléter le document réponse (1) en plaçant  $\mathcal{R}_A$ ,  $\mathcal{R}_B$ ,  $r$ ,  $NI$ ,  $\varphi_A$  et  $\varphi_B$  les flux magnétiques traversant les entrefers aux points A et B.

**1. 1. 5.** Déterminer l'expression des flux  $\varphi_A$  et  $\varphi_B$  en fonction de  $N$ ,  $I$ ,  $\mu_0$ ,  $S$ ,  $e$  et  $r$ .

Chaque entrefer est commun à deux circuits élémentaires.

**1. 1. 6.** Exprimer le flux magnétique total  $\phi_A$  qui traverse l'entrefer en A.

**1. 1. 7.** En déduire les expressions des flux magnétiques  $\phi_B$ ,  $\phi_C$ ,  $\phi_D$ ,  $\phi_E$ ,  $\phi_F$ ,  $\phi_G$  et  $\phi_H$  qui traversent les entrefers B, C, D, E, F, G, H.

**1. 1. 8.** Exprimer l'amplitude du champ magnétique  $B_{pol}$  dans les différents entrefers en fonction de  $\mu_0$ ,  $N$ ,  $I$ ,  $e$ ,  $r$ , et  $\mathcal{R}_0 = \frac{e}{\mu_0 S}$ .

## 1.2 Bobinages d'excitation et de détection

Sur le même circuit magnétique (partie stator), deux bobinages d'excitation sont mis en place afin de pouvoir mettre en vibration le cylindre. Les bobinages sont disposés comme indiqué sur les figures (8) et (9). Ils se composent respectivement de  $n_1$  spires/branche pour le bobinage 1 et de  $n_2$  spires/branche pour le bobinage 2.

Pour des raisons de simplicité et de couplage magnétique, les deux bobinages d'excitation sont également utilisés pour la détection de la vibration du cylindre. En effet un dispositif de séquençage permet d'utiliser les bobinages soit en excitation soit en détection.

Nous supposons que la vibration du cylindre a lieu selon l'axe  $AE$  (également nommé *axe 1*). Ceci induit une modification de l'entrefer au niveau de chaque branche caractérisée par :

$$e_A = e_E = e + v_1(t)$$

$$e_C = e_G = e - v_1(t)$$

$$e_B = e_D = e_F = e_H = e$$

Nous conservons les hypothèses faites pour la partie 1.1 (pas de flux de fuite et le cylindre est considéré parfait).

Toujours pour des raisons de symétrie, le circuit magnétique total peut se décomposer en quatre circuits magnétiques élémentaires comme indiqué par la figure (10).

**1. 2. 1.** A partir du théorème d'Ampère, déterminer les ré reluctances  $\mathcal{R}_A$ ,  $\mathcal{R}_B$ ,  $\mathcal{R}_C$  et  $r$  respectivement des entrefers  $A$ ,  $B$ ,  $C$  et du cylindre. On prendra  $S_1 = S$ .

**1. 2. 2.** En vous inspirant du schéma fourni par le document réponse (1), donner le schéma électrique équivalent du circuit magnétique élémentaire décrit par la figure (10). Vous placerez sur le schéma  $\mathcal{R}_A$ ,  $\mathcal{R}_B$ ,  $\mathcal{R}_C$ ,  $r$ ,  $NI$ ,  $\varphi_A$ ,  $\varphi_B$  et  $\varphi_C$  les flux magnétiques qui traversent les entrefers aux points  $A$ ,  $B$  et  $C$ .

**1. 2. 3.** Calculer l'expression de  $\varphi_A$  le flux magnétique de l'entrefer au niveau du point  $A$ . Le flux sera exprimé en fonction  $N$ ,  $I$ ,  $\mathcal{R}_A$ ,  $\mathcal{R}_B$ ,  $\mathcal{R}_C$  et  $r$ .

**1. 2. 4.** Exprimer  $\varphi_A$  en fonction de  $N$ ,  $I$ ,  $e$ ,  $r$ ,  $\mathcal{R}_0$  et  $v_1(t)$ . Nous négligerons le terme  $\frac{v_1^2(t)}{e^2}$ .

**1. 2. 5.** Calculer l'expression de  $\varphi_C$  le flux magnétique de l'entrefer au niveau du point  $C$ . Le flux sera exprimé en fonction  $N$ ,  $I$ ,  $\mathcal{R}_A$ ,  $\mathcal{R}_B$ ,  $\mathcal{R}_C$  et  $r$ .

**1. 2. 6.** Exprimer  $\varphi_C$  en fonction de  $N$ ,  $I$ ,  $e$ ,  $r$ ,  $\mathcal{R}_0$  et  $v_1(t)$ . Nous négligerons le terme  $\frac{v_1^2(t)}{e^2}$ .

**1. 2. 7.** Calculer l'expression de  $\varphi_B$  le flux magnétique de l'entrefer au niveau du point  $B$ . Le flux sera exprimé en fonction  $N$ ,  $I$ ,  $\mathcal{R}_A$ ,  $\mathcal{R}_B$ ,  $\mathcal{R}_C$  et  $r$ .

**1. 2. 8.** Exprimer  $\varphi_B$  en fonction de  $N$ ,  $I$ ,  $e$ ,  $r$ ,  $\mathcal{R}_0$  et  $v_1(t)$ . Nous négligerons le terme  $\frac{v_1^2(t)}{e^2}$ .

**1. 2. 9.** En considérant le circuit magnétique total, exprimer les flux magnétiques  $\phi_A$ ,  $\phi_B$  et  $\phi_C$  qui traversent respectivement les entrefers  $A$ ,  $B$  et  $C$ . (Pour guider le candidat on précise que  $\phi_A$  représente le flux magnétique total au niveau de l'entrefer  $A$ , il s'exprime par  $\phi_A = 2.\varphi_A$ )

Les vibrations font varier l'entrefer et par conséquent font varier le flux magnétique. Le bobinage 1, décrit par la figure (8) et constitué de  $n_1$  spires par branche, détecte la variation de flux.

**1. 2. 10.** Donner l'expression de la tension induite sur la branche  $A$ , notée  $U_A$ , dans le bobinage 1.  $U_A$  sera exprimée en fonction de  $N$ ,  $n_1$ ,  $I$ ,  $e$ ,  $r$ ,  $\mathcal{R}_0 = \frac{e}{\mu_0.S}$  et  $v_1(t)$ .

**1. 2. 11.** Donner l'expression de la tension induite sur la branche  $C$ , notée  $U_C$ , dans le bobinage 1.

**1. 2. 12.** A partir des deux questions précédentes, exprimer  $U_1$  la tension totale induite dans la bobinage 1 lorsque le cylindre vibre selon l'axe  $AE$ . (Le bobinage est tel que  $U_1 = U_A + U_E - U_C - U_G$ )

**1. 2. 13.** Montrer que  $U_1 = G_{det1} \cdot \frac{dv_1(t)}{dt}$ . Exprimer  $G_{det1}$ .

Pour le bobinage 2, nous allons considérer une vibration du cylindre selon l'axe  $HD$  (voir la figure (9)). Le bobinage 2 comporte  $n_2$  spires par branche et la vibration sera notée  $v_2(t)$ .

**1. 2. 14.** Exprimer  $e_A, e_B, e_C, e_D, e_E, e_F, e_G, e_H$  en fonction de  $e$  et de  $v_2(t)$ .

**1. 2. 15.** Montrer que  $U_2 = G_{det2} \cdot \frac{dv_2(t)}{dt}$ . Exprimer  $G_{det2}$ .

En conclusion le système de détection sera modélisé par :

$$\begin{pmatrix} U_1(t) \\ U_2(t) \end{pmatrix} = \begin{bmatrix} G_{det1} & 0 \\ 0 & G_{det2} \end{bmatrix} \cdot \begin{pmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{pmatrix}$$

La partie excitation n'est pas étudiée dans le devoir, son modèle est donné par :

$$\begin{pmatrix} F_{mag1}(t) \\ F_{mag2}(t) \end{pmatrix} = \begin{bmatrix} G_{exc1} & 0 \\ 0 & G_{exc2} \end{bmatrix} \cdot \begin{pmatrix} i_1(t) \\ i_2(t) \end{pmatrix} + \begin{bmatrix} K_{mag1} & 0 \\ 0 & K_{mag2} \end{bmatrix} \cdot \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix}$$

où  $i_1(t)$  et  $i_2(t)$  sont les courants qui circulent dans les bobinages 1 et 2 et  $F_{mag1}(t)$  et  $F_{mag2}(t)$  sont les forces magnétiques qui font vibrer le cylindre.

### 1.3 Modélisation du cylindre

Le cylindre est excité par des forces magnétiques notées  $F_{mag1}(t)$  et  $F_{mag2}(t)$  selon qu'elles proviennent du bobinage 1 ou du bobinage 2. Nous considéreront qu'elles sont appliquées à une hauteur  $x_0$  comme indiqué sur la figure (11).

A partir de la structure de l'ensemble, il est possible de montrer que le comportement dynamique du cylindre peut être modélisé par la relation mathématique suivante (dans le cas d'un cylindre parfait) :

$$\begin{pmatrix} F_{mag1}(t) \cdot f^2(X_0) \\ F_{mag2}(t) \cdot f^2(X_0) \end{pmatrix} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \cdot \begin{pmatrix} \ddot{v}_1(t) \\ \ddot{v}_2(t) \end{pmatrix} + \begin{bmatrix} \frac{M\omega_0}{Q} & 0 \\ 0 & \frac{M\omega_0}{Q} \end{bmatrix} \cdot \begin{pmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{pmatrix} + \begin{bmatrix} M\omega_0^2 & 0 \\ 0 & M\omega_0^2 \end{bmatrix} \cdot \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix} + \begin{bmatrix} 0 & 2k_c\Omega(t) \\ 2k_c\Omega(t) & 0 \end{bmatrix} \cdot \begin{pmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{pmatrix}$$

**1. 3. 1.** Dans le cas où les deux axes ne sont pas couplés ( $\Omega(t) = 0 \text{ rad.s}^{-1}$ ), exprimer l'équation différentielle qui lie les grandeurs  $v_1(t)$  et  $i_1(t)$  en fonction de  $G_{exc1}$ ,  $K_{mag1}$ ,  $M$ ,  $\omega_0$ ,  $Q$  et  $f(X_0)$ .

**1. 3. 2.** Dans le cas où  $\Omega(t) = 0 \text{ rad.s}^{-1}$ , calculer la fonction de transfert  $G_1(p) = \frac{V_1(p)}{I_1(p)}$ .

**1. 3. 3.** Lorsque le terme  $K_{mag1} \cdot f^2(X_0)$  est nul, calculer la pulsation de résonance,  $\omega_r$ , du cylindre (en régime harmonique) en fonction de  $\omega_0$  la pulsation naturelle et  $Q$  le coefficient de qualité du cylindre.

**1. 3. 4.** Sur le document réponse 2, compléter la réponse fréquentielle de Bode en positionnant notamment les pulsations  $\omega_0$  et  $\omega_r$ .

**1. 3. 5.** Le terme  $K_{mag1} \cdot f^2(X_0)$  n'est plus considéré nul, calculer la nouvelle pulsation de résonance,  $\omega'_r$ , du cylindre (en régime harmonique).

## 1.4 Modélisation de l'ensemble Excitation-Cylindre vibrant-Détection

Le modèle du système peut être représenté comme indiqué sur la figure (12).

La partie *excitation* (non étudiée dans ce devoir) est caractérisée par :

$$\begin{pmatrix} F_{mag1}(t) \\ F_{mag2}(t) \end{pmatrix} = \begin{bmatrix} G_{exc1} & 0 \\ 0 & G_{exc2} \end{bmatrix} \cdot \begin{pmatrix} i_1(t) \\ i_2(t) \end{pmatrix} + \begin{bmatrix} K_{mag1} & 0 \\ 0 & K_{mag2} \end{bmatrix} \cdot \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix}$$

Le signal d'entrée pour le cylindre est constitué des forces  $F_{mag1}$  et  $F_{mag2}$ .

Le vecteur d'état est constitué des signaux :

$$x = \begin{pmatrix} v_1 \\ \dot{v}_1 \\ v_2 \\ \dot{v}_2 \end{pmatrix}$$

**1. 4. 1.** Calculer la représentation d'état du cylindre vibrant et de la détection, en donnant l'expression des matrices  $A$ ,  $B$ ,  $C$  et  $D$ .

$$\begin{cases} \dot{x} = A.x + B.u \\ y = C.x + D.u \end{cases}$$

avec

$$y = \begin{pmatrix} U_1(t) \\ U_2(t) \end{pmatrix}$$

**1. 4. 2.** Exprimer la représentation d'état de l'ensemble Excitation-Cylindre vibrant-Détection. Les vecteurs  $x$  et  $y$  sont les mêmes que pour la question précédente. Le vecteur de commande est défini par :

$$u = \begin{pmatrix} i_1(t) \\ i_2(t) \end{pmatrix}$$

**1. 4. 3.** Le modèle obtenu est-il linéaire ? Justifier rapidement la réponse.

## 1.5 Modélisation du séquençage

Les systèmes d'excitation et de détection utilisent les mêmes bobines. Un système de séquençage permet soit d'exciter soit de détecter, de telle sorte que les signaux d'excitation et les signaux de détection ne peuvent coexister simultanément. Ce dispositif est décrit sur la figure (13).

Lorsque  $\alpha(t) = 1$  les interrupteurs pour l'excitation sont fermés.

Lorsque  $\overline{\alpha}(t) = 1$  les interrupteurs pour la détection sont fermés.

Ces deux signaux sont périodiques, la période est notée  $T$ .

**1. 5. 1.** Calculer les coefficients  $C_n$  de la série de Fourier de  $\alpha(t)$ .

$$\alpha(t) = \sum_{n=-\infty}^{+\infty} C_n \cdot e^{\frac{2\pi j n}{T} t} \text{ avec } C_n = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} \alpha(t) \cdot e^{-\frac{2\pi j n}{T} t} \cdot dt$$

**1. 5. 2.** Calculer les coefficients  $\overline{C}_n$  de la série de Fourier de  $\overline{\alpha}(t)$ .

$$\overline{\alpha}(t) = \sum_{n=-\infty}^{+\infty} \overline{C}_n \cdot e^{\frac{2\pi j n}{T} t}$$

**1. 5. 3.** Sur le document réponse 3, représenter le spectre d'amplitude ( $|\alpha(f)|$ ) de  $\alpha(t)$ . Vous préciserez entre autres les valeurs pour lesquelles  $|\alpha(f)| = 0$ .

**1. 5. 4.** Sur le document réponse 4, représenter le spectre d'amplitude ( $|\overline{\alpha}(f)|$ ) de  $\overline{\alpha}(t)$ . Vous préciserez entre autres les valeurs pour lesquelles  $|\overline{\alpha}(f)| = 0$ .

**1. 5. 5.** Exprimer  $i_1(t)$  en fonction de  $\alpha(t)$  et de  $i_1'(t)$ .

**1. 5. 6.** Exprimer  $I_1(f)$  en fonction de  $\alpha(f)$  et de  $I_1'(f)$ . ( $I_1(f) = TF[i_1(t)]$  et  $I_1'(f) = TF[i_1'(t)]$ )

**1. 5. 7.**  $i_1'(t) = I_{10}' \cdot \sin(\omega_0 t)$ . Calculer  $I_1'(f)$  la transformée de Fourier de  $i_1'(t)$ . On rappelle la définition de la transformée de Fourier d'un signal :  $I_1'(f) = TF[i_1'(t)] = \int_{-\infty}^{+\infty} i_1'(t) \cdot e^{-2\pi j f t} \cdot dt$ . On peut également s'aider des relations suivantes :  $\sin(\omega_0 t) = \frac{e^{j\omega_0 t} - e^{-j\omega_0 t}}{2j}$  et  $TF[e^{-2\pi j f_0 t}] = \delta(f - f_0)$ .

**1. 5. 8.** Sur le document réponse 5, tracer le spectre d'amplitude de  $i_1'(t)$ .

**1. 5. 9.** Sur le document réponse 6, tracer le spectre d'amplitude de  $i_1(t)$ .

**1. 5. 10.** Le signal  $i_1(t)$  passe à travers le gyromètre qui est un système très résonant, est-il nécessaire de considérer l'ensemble du spectre de  $i_1(t)$  pour établir le modèle de l'ensemble ? Justifier rapidement votre réponse.

**1. 5. 11.** Proposer un modèle pour la partie **excitation** en intégrant le système de séquençage.



## PARTIE 2 Commande du gyromètre

Nous proposons une étude d'une partie du système de régulation. Le modèle utilisé est donné sous la forme d'une représentation d'état.

$x$  : vecteur d'état ;

$u$  : vecteur de commande ;

$y$  : vecteur de sortie.

$$x = \begin{pmatrix} v_1 & \dot{v}_1 & v_2 & \dot{v}_2 \end{pmatrix}^T$$

$$u = \begin{pmatrix} i_1 & i_2 \end{pmatrix}^T$$

$$y = \begin{pmatrix} u_1 & u_2 \end{pmatrix}^T$$

$$\begin{cases} \dot{x} = \begin{bmatrix} A_1 & \mathbb{O} \\ \mathbb{O} & A_1 \end{bmatrix} .x + \Omega(t) \begin{bmatrix} \mathbb{O} & -A_2 \\ A_2 & \mathbb{O} \end{bmatrix} .x + \frac{\tau}{T} \cdot \begin{bmatrix} B_1 & \mathbb{O} \\ \mathbb{O} & B_1 \end{bmatrix} .u \\ y = \frac{T-\tau}{T} \begin{bmatrix} C_1 & \mathbb{O} \\ \mathbb{O} & C_1 \end{bmatrix} .x \end{cases}$$

où

$$A_1 = \begin{bmatrix} -\frac{A}{M} & -\frac{K}{M} \\ 1 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} -\frac{2M_c}{M} & 0 \\ 0 & 0 \end{bmatrix}$$
$$B_1 = \begin{bmatrix} \frac{G_{exc} \cdot K_z^2}{M} \\ 0 \end{bmatrix} \quad C_1 = \begin{bmatrix} G_{det} & 0 \end{bmatrix}$$

et  $\mathbb{O}$  : matrices de dimension convenable pour lesquelles tous ses éléments sont nuls.

### 2.1 Propriétés du modèle

Le modèle donné ci-dessus est destiné au calcul d'un ensemble de correcteurs. La vitesse de rotation du cylindre autour de son axe est envisagée comme une perturbation. L'objectif est de maintenir l'ellipse de vibration sur l'axe 1.

**2. 1. 1.** Quel(s) avantage(s) peut présenter l'écriture du modèle sous la forme d'une représentation d'état plutôt qu'une matrice de transfert ?

**2. 1. 2.** Afin de vérifier les propriétés de commandabilité, quelle condition doit-être vérifiée pour que le modèle soit commandable ? On notera  $M_C$  la matrice de commandabilité. La condition sera exprimée en fonction de  $A_1$  et  $B_1$ . (Nous considérerons  $\Omega(t) = 0 \text{ rad.s}^{-1}$ )

**2. 1. 3.** Appliquer la condition donnée à la question précédente et conclure sur la commandabilité du modèle. (On se limitera aux six premières colonnes de la matrice  $M_C$ )

**2. 1. 4.** Quelle propriété doit vérifier le modèle pour être observable ? On notera  $M_O$  la matrice d'observabilité. La condition sera exprimée en fonction de  $A_1$  et  $C_1$ . (Nous considérerons  $\Omega(t) = 0 \text{ rad.s}^{-1}$ )

**2. 1. 5.** Pour  $\Omega(t) = 0 \text{ rad.s}^{-1}$ , le modèle est-il observable ? (Pour le calcul, on se limitera aux six premières lignes de la matrice  $M_O$ )

## 2.2 Etude de la stabilité du modèle

**2. 2. 1.** Pour  $\Omega(t) = \Omega_0$ , exprimer la relation qui permet de calculer la matrice de transfert du modèle.

La matrice de transfert du modèle, pour  $\Omega_0 = 0 \text{ rad.s}^{-1}$ , est donnée par :

$$G(p) = \begin{bmatrix} \frac{N(p)}{D(p)} & 0 \\ 0 & \frac{N(p)}{D(p)} \end{bmatrix}$$

avec

$$D(p) = p^2 + 11,6688.p + 2,4019.10^9;$$

$$N(p) = 2,8067.p - 9,5367.10^{-7}.$$

**2. 2. 2.** Le modèle décrit par  $G(p)$  est-il stable ? Justifier votre réponse par le calcul.

Une seconde approche consiste à caractériser la stabilité à partir de la représentation d'état.

**2. 2. 3.** Quelle propriété doit avoir la matrice d'état  $A$  pour que le modèle soit stable ?

$$A = \begin{bmatrix} A_1 & \Omega_0 A_2 \\ -A_2 \Omega_0 & A_1 \end{bmatrix}$$

**2. 2. 4.** Quelle relation permet de calculer les valeurs propres d'une matrice  $A$  ?

**2. 2. 5.** Quelle(s) signification(s) donnez vous à la notion de valeur propre ?

**2. 2. 6.** Calculer les valeurs propres du modèle pour  $\Omega_0 = 0 \text{ rad.s}^{-1}$ . Pour calculer les valeurs propres de  $A$  il suffit de calculer celles de  $A_1$ .

**2. 2. 7.** Le modèle est-il stable ?

Une troisième approche est celle proposée par Lyapunov. L'idée de départ est de choisir une fonction notée  $V(x)$  ( $x$  étant le vecteur d'état). La fonction  $V(x)$  peut, par exemple, représenter l'énergie du vecteur d'état. Si cette énergie est décroissante, en régime libre, au cours du temps alors le système sera qualifié de stable. Mathématiquement cela se résume par :

$$\begin{cases} V(x) &= x^T \cdot x \\ \dot{V}(x) &= \frac{dV(x)}{dt} < 0 \end{cases} \quad (1)$$

La relation donnée ci dessus est une condition suffisante de stabilité, c'est à dire que lorsqu'elle est satisfaite on peut conclure à la stabilité du système étudié. Dans le cas où elle n'est pas satisfaite, on ne peut pas conclure sur la stabilité du système : il peut être stable ou instable.

La stabilité est une propriété intrinsèque du système. Pour la caractériser, on se limitera à la partie du modèle :

$$\dot{x} = A.x$$

avec  $\Omega_0 = 0 \text{ rad.s}^{-1}$ .

**2. 2. 8.** Calculer  $\dot{V}(x) = \frac{dV(x)}{dt}$  en fonction de  $A$  et de  $x$ . Le résultat pour  $\dot{V}(x)$  sera donné sous la forme  $x^T \cdot (\dots) \cdot x$ .

**2. 2. 9.** De l'inégalité  $x^T \cdot F \cdot x < 0$  on obtient  $F < 0$ . A partir de la question précédente exprimer  $F$  en fonction de  $A_1$  ( $\Omega_0 = 0 \text{ rad.s}^{-1}$ ).

**2. 2. 10.** L'inégalité  $F < 0$  est vérifiée si et seulement si toutes les valeurs propres de  $F$  présentent une partie réelle strictement négative. Calculer les valeurs propres de  $F$ .

**2. 2. 11.** A partir de la question précédente que peut on dire quant à la stabilité du système ?

Il est possible de généraliser la fonction  $V(x)$  comme :

$$V(x) = x^T \cdot P \cdot x \quad (2)$$

où  $P$  est une matrice symétrique ( $P = P^T$ ) telle que toutes ses valeurs propres aient une partie réelle strictement positive.

**2. 2. 12.** Calculer  $\dot{V}(x) = \frac{dV(x)}{dt}$  en fonction de  $A$ ,  $P$  et  $x$ .

Pour résoudre l'inégalité matricielle  $F' < 0$ , on la transforme en égalité matricielle  $F' + Q = 0$ . Afin de simplifier les calculs,  $Q$  est par exemple une matrice identité :

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

**2. 2. 13.** Calculer les valeurs propres de la matrice  $Q$ .

**2. 2. 14.** Quelle doit être l'ordre de la matrice  $P$  ?

**2. 2. 15.** Calculer la matrice  $P$  (matrice symétrique) qui vérifie l'égalité :  $F' + Q = 0$ .

**2. 2. 16.** Calculer les valeurs propres de la matrice  $P$ .

**2. 2. 17.** Le modèle proposé est-il stable ?

**2. 2. 18.** Quel est l'intérêt d'introduire une matrice  $P$  dans la fonction  $V(x)$  ?

## PARTIE 3 Implantation du dispositif de commande dans un composant programmable

Nous n'aborderons pas l'implantation de l'ensemble du dispositif de commande mais seulement les éléments les plus délicats telle la boucle à verrouillage de phase.

### 3.1 Algorithme de CORDIC

La boucle à verrouillage de phase proposée utilise un algorithme nommé CORDIC (COordinate Rotation DIgital Computer). Il est utilisé notamment pour l'implantation de fonctions trigonométriques. Nous allons l'étudier dans le cas de l'implantation d'une fonction *sinus*.

Il s'agit d'une méthode itérative qui nécessite uniquement la multiplication par deux, donc particulièrement adaptée aux composants programmables. Son symbole est le suivant :

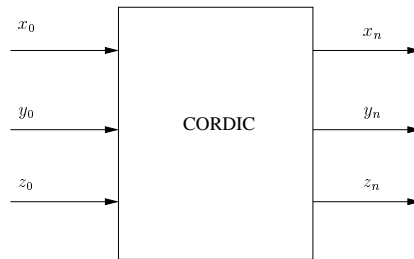


Fig. 3 Symbole du composant CORDIC

Les différentes itérations de l'algorithme sont données par :

$$\begin{cases} x_{i+1} &= x_i - \mu_i \cdot y_i \cdot \delta_i \\ y_{i+1} &= y_i + \mu_i \cdot x_i \cdot \delta_i \\ z_{i+1} &= z_i - \mu_i \cdot \alpha_i \end{cases} \quad (4)$$

**3. 1. 1.** Ecrire les deux premières équations sous la forme matricielle en calculant la matrice  $C_i$ .

$$v_{i+1} = C_i \cdot v_i$$

avec  $v_i = (x_i, y_i)^T$  et  $v_{i+1} = (x_{i+1}, y_{i+1})^T$ .

La matrice  $C_i$  peut également s'écrire :  $C_i = K_i \cdot R_i$ .

$K_i$  est un scalaire correspondant à un facteur d'échelle et  $R_i$  est une matrice de rotation de même dimension de  $C_i$  et pouvant s'écrire :

$$R_i = \begin{bmatrix} \cos(\alpha_i) & -\mu_i \cdot \sin(\alpha_i) \\ \mu_i \cdot \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \quad (5)$$

$\mu_i = \pm 1$  et détermine le sens de rotation.

**3. 1. 2.**  $C_i = K_i \cdot R_i$ , calculer  $K_i$  et  $\delta_i$  en fonction de  $\alpha_i$ .

**3. 1. 3.** Soit  $v_0$  la valeur initiale de  $v_i$ , exprimer le vecteur  $v_i$  en fonction de  $v_0$  et de  $C_{i-1}, C_{i-2}, \dots$ .

**3. 1. 4.** Exprimer  $v_n$  (vecteur  $v_i$  après  $n$  itérations) en fonction de  $v_0, K_{n-1}, K_{n-2}, \dots$  et  $R_{n-1}, R_{n-2}, \dots$ .

Nous considérons  $\delta_i = 2^{-i}$  avec  $i = 0, 1, 2, \dots, n$ .

**3. 1. 5.** Calculer  $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{10}$ .

Sachant que l'angle de rotation (après  $n$  itérations) s'exprime par :

$$\alpha = \sum_{i=0}^n \mu_i \cdot \alpha_i$$

**3. 1. 6.** Dans le cas particulier où le sens de rotation est positif ( $\mu_i = 1$ ), calculer la valeur de  $\alpha$  pour  $n = 10$ .

Le nombre  $K$  est appelé *facteur d'échelle*. Il a pour expression :

$$K = K_0 \cdot K_1 \cdot K_2 \cdot \dots \cdot K_n = \prod_{i=0}^n K_i$$

**3. 1. 7.** Exprimer  $K_i$  en fonction de  $\delta_i$ .

**3. 1. 8.** Nous considérons  $\delta_i = 2^{-i}$ . Calculer  $K$  pour  $n = 10$ .

La troisième équation de l'algorithme de CORDIC est égale à :

$$z_{i+1} = z_i - \mu_i \cdot \alpha_i$$

**3. 1. 9.** Exprimer  $z_n$ , valeur de  $z$  après  $n$  itérations, en fonction de  $z_0, \mu_i, \alpha_i$  et  $n$ .

On définit l'angle de rotation à la  $i^{me}$  itération :

$$angle = \sum_{k=0}^i \mu_k \cdot \alpha_k$$

$$\begin{cases} \mu_{i+1} = +1 & \text{lorsque } angle < z_0 \\ \mu_{i+1} = -1 & \text{lorsque } angle > z_0 \end{cases} \quad (6)$$

**3. 1. 10.** Vers quelle valeur tend  $z_n$  lorsque  $n$  est très grand ?

**3. 1. 11.** On envisage le calcul du sinus d'un angle de  $53^\circ$  ( $z_0 = 53^\circ$ ). L'algorithme étudié dans cette partie est envisagé avec 10 itérations ( $i = 0, 1, 2, \dots, 9, 10$ ). Sur le document réponse 7 compléter le tableau.

Après  $n$  itérations, l'algorithme fournit :

$$v_n = K \cdot \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot v_0 \quad (7)$$

- 3. 1. 12.** Dans quelle colonne du document réponse 7 obtient-on la valeur de  $\sin(53^\circ)$  ? Justifier rapidement votre réponse.
- 3. 1. 13.** Quelle opération simple reste-t-il à effectuer pour obtenir la valeur correcte de  $\sin(53^\circ)$  ?
- 3. 1. 14.** Les résultats obtenus sur le document réponse 7 permettent-ils d'obtenir la valeur du cosinus de  $53^\circ$  ? Justifier rapidement votre réponse.
- 3. 1. 15.** Après 10 itérations ( $i = 10$ ), quelle erreur commet-on sur le calcul du sinus par rapport à la valeur théorique ?
- 3. 1. 16.** Dans l'hypothèse où le sinus serait précalculé sur un quart de période, combien de bits seraient nécessaires au codage de ses valeurs pour avoir une erreur plus faible que celle obtenue avec l'algorithme de CORDIC à la question précédente ?
- 3. 1. 17.** Quel serait l'encombrement "mémoire" pour stocker les valeurs précalculées sur le quart de période (l'argument du sinus est codé avec 12 bits) ?
- 3. 1. 18.** D'après la documentation du composant XC2S200 (XILINX), est-il possible de stocker les valeurs d'un sinus précalculé sur un quart de période avec une précision suffisante dans la mémoire RAM du composant ? (voir Annexe 1)
- 3. 1. 19.** Quel(s) intérêt(s) peut présenter l'algorithme de CORDIC pour le calcul d'une fonction sinus ?

## 3.2 Boucle à verrouillage de phase

Pour réaliser la boucle à verrouillage de phase (PLL), la cible est un composant programmable de type FPGA. Le schéma général de boucle à verrouillage de phase est le suivant :

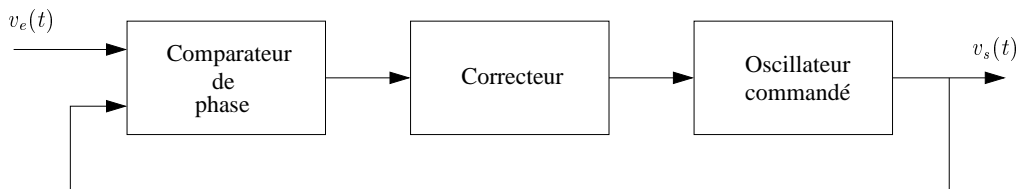


Fig. 4 Schéma général d'une PLL

### 3.2.1 Comparateur de phase

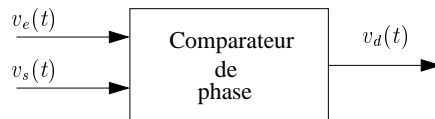


Fig. 5 Comparateur de phase

$$v_e(t) = V_e \cdot \sin(\phi_e(t))$$

$$v_s(t) = V_s \cdot \sin(\phi_s(t))$$

Nous cherchons à établir la sortie du comparateur de phase telle que :

$$v_d(t) = K_d \cdot (\phi_e(t) - \phi_s(t))$$

### Comparateur à multiplieur

**3. 2. 1. 1.** Calculer le produit  $v_e(t).v_s(t)$  en mettant en évidence les deux termes :  $\cos(\phi_e(t) - \phi_s(t))$  et  $\cos(\phi_e(t) + \phi_s(t))$ .

**3. 2. 1. 2.** Dans le cas où  $v_e(t)$  et  $v_s(t)$  sont à des fréquences voisines, exprimer les fréquences des deux termes mis en évidence dans la question précédente.

**3. 2. 1. 3.** Pour réaliser le comparateur de phase, lequel des deux termes faut-il éliminer ? Quelle fonction permet cette élimination ?

**3. 2. 1. 4.** Exprimer  $v_d(t)$  après élimination du terme précisé dans la question précédente.

**3. 2. 1. 5.** Sur le document réponse 8, tracer graphiquement  $v_d$  en fonction de  $\phi_e(t) - \phi_s(t)$ .

**3. 2. 1. 6.** Autour de quel point est-il intéressant de faire fonctionner le comparateur de phase pour obtenir la relation :  $v_d \simeq K_d.(\phi_s(t) - \phi_e(t))$  ?

**3. 2. 1. 7.** Donner l'expression de  $K_d$  en fonction de  $V_e$  et de  $V_s$ .

**3. 2. 1. 8.** Quelles sont les différentes fonctions à implanter dans le composant programmable ?

### Comparateur : algorithme de CORDIC

Afin d'éviter l'implantation des multiplications dans le composant programmable on utilise l'algorithme de CORDIC étudié dans la partie **3.1**.

Après  $n$  itérations, l'algorithme converge vers les relations suivantes :

$$\begin{cases} x_n &= K.(x_0.\cos(z_0) - y_0.\sin(z_0)) \\ y_n &= K.(y_0.\cos(z_0) + x_0.\sin(z_0)) \\ z_n &= 0 \end{cases} \quad (8)$$

**3. 2. 1. 9.** Préciser les valeurs d'initialisation  $x_0$ ,  $y_0$  et  $z_0$  pour réaliser le comparateur de phase.

**3. 2. 1. 10.** La fonction obtenue à la question précédente comporte également deux termes similaires à ceux évoqués à la question 3.2.1.1. Après élimination d'un des deux termes et linéarisation du terme restant, quelle est la valeur de  $K_d$  qui modélise le comparateur de phase ?

### **3.2.2 Oscillateur commandé**

L'oscillateur est entièrement numérique. Par analogie avec le cas **analogique** où l'oscillateur est piloté par une tension, l'oscillateur **numérique** est piloté par un nombre binaire noté  $N$ . La fréquence de sortie de l'oscillateur est alors proportionnelle à  $N$ .

$$f_0 = K_0.N$$

La réalisation de l'oscillateur numérique comporte deux parties : un accumulateur de phase (intégrateur numérique) et un CORDIC. L'accumulateur de phase possède la fonction de transfert :

$$I(Z) = \frac{1}{1 - Z^{-1}}.$$

La fréquence d'échantillonnage est donnée par la fréquence d'horloge  $F_{clk}$  (période  $T_{clk}$ ). Le nombre de bits de l'accumulateur est noté  $n_a$ .

**3. 2. 2. 1.** *Rappeler la définition de  $Z$ , variable de la transformée en  $Z$ .*

**3. 2. 2. 2.** *Pour un nombre  $N$  constant et égal à 1, donner sur le document réponse 9 la courbe représentant la sortie de l'accumulateur pour les 10 premiers échantillons.*

**3. 2. 2. 3.** *Quelle est la valeur maximale de sortie de l'accumulateur ?*

**3. 2. 2. 4.** *Quelle est la fréquence du signal de sortie de l'accumulateur ( $f_{sa}$ ) ? On exprimera  $f_{sa}$  en fonction de  $F_{clk}$ ,  $N$  et  $n_a$ .*

**3. 2. 2. 5.** *Calculer l'expression  $\phi(t)$  de la phase (sortie de l'accumulateur de phase) en fonction de  $F_{clk}$ ,  $n_a$  et  $N$ .*

Le signal de sortie de l'accumulateur représente une phase.

**3. 2. 2. 6.** *Compléter le document réponse 10 (zones grisées) pour obtenir les sorties indiquées sur le même document.*

**3. 2. 2. 7.** *Calculer la valeur de  $K_0$  du gain de l'accumulateur de phase.*

La boucle à verrouillage de phase peut être représentée par la figure du document réponse .

On peut remarquer que  $\omega_0$  sert à régler la pulsation centrale de la boucle à verrouillage de phase.

**3. 2. 2. 8.** *Compléter le document réponse 11 en indiquant ce qui se trouve dans les parties grisées.*

**3. 2. 2. 9.** *On modélise la boucle à verrouillage de phase par des schémas blocs. Compléter le document réponse 12 en indiquant les modèles dans les différents blocs vides.*

**3. 2. 2. 10.** *Quel correcteur faut-il prévoir pour avoir une erreur de phase et de fréquence nulle ? Justifier votre réponse.*



# FIGURES DE LA PARTIE 1

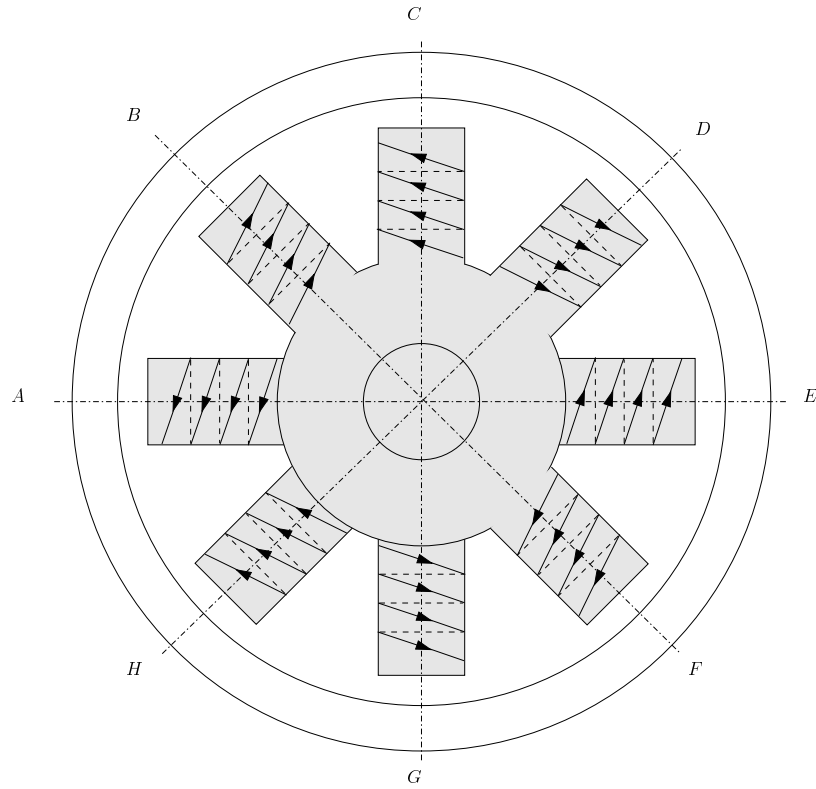


Fig. 6 Bobinage de polarisation

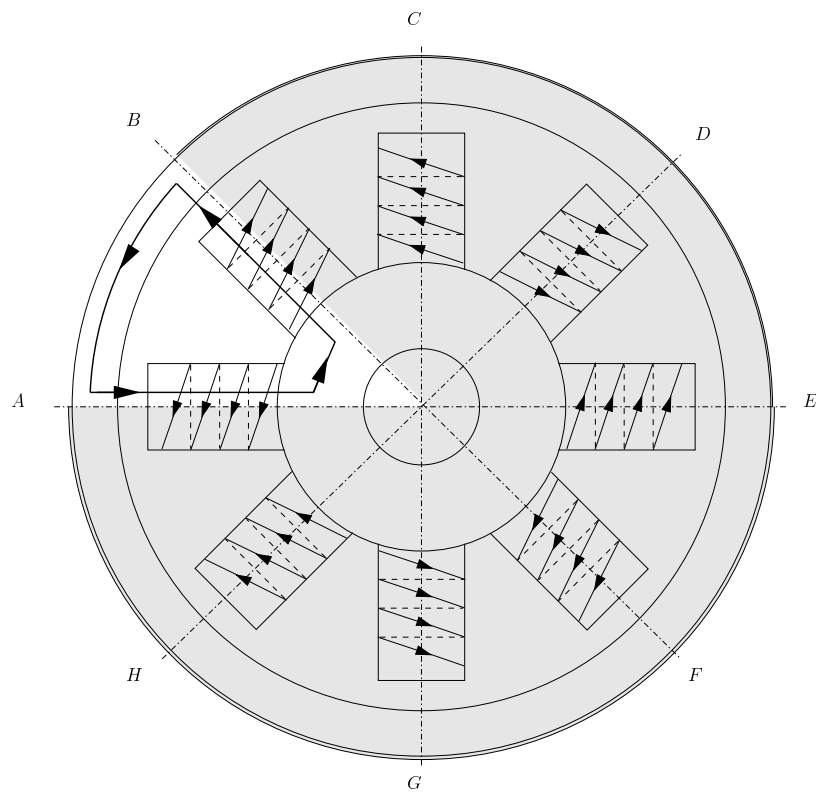


Fig. 7 Bobinage de polarisation élémentaire

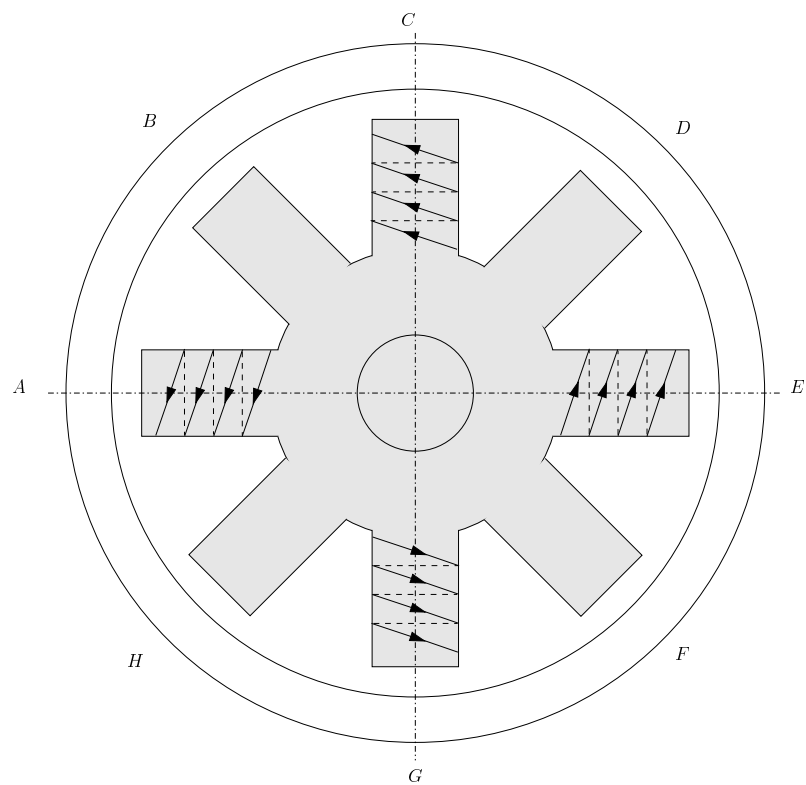


Fig. 8 Bobinage 1 (excitation et détection)

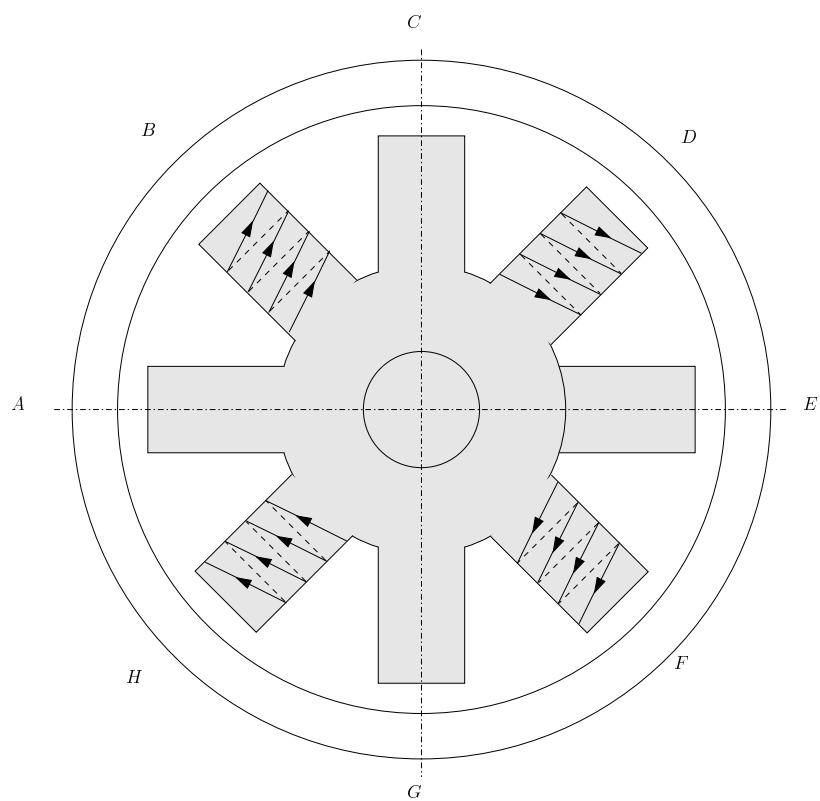


Fig. 9 Bobinage 2 (excitation et détection)

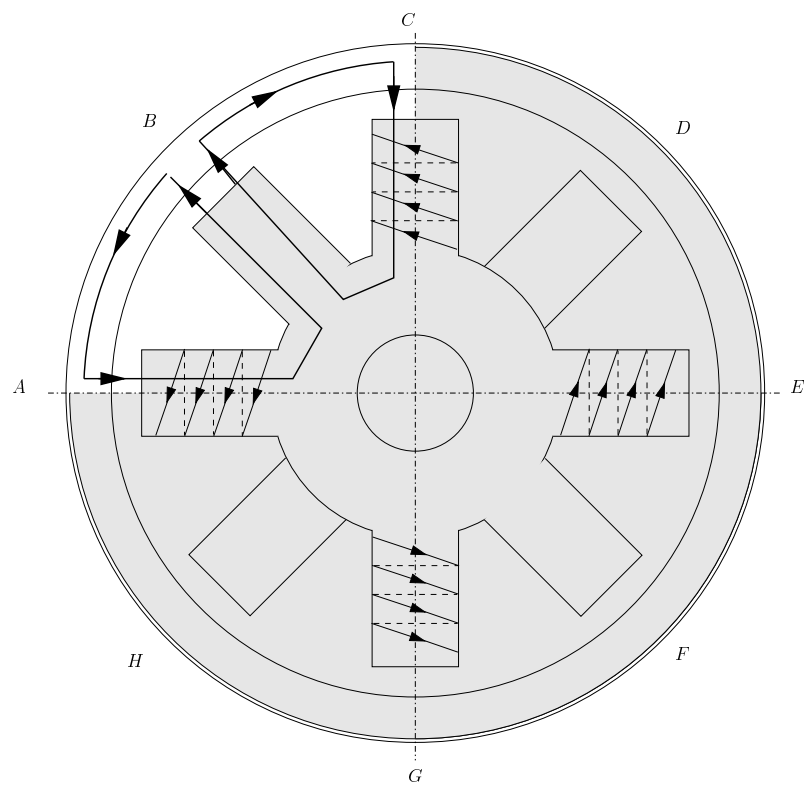


Fig. 10 Circuit magnétique élémentaire : détection par le bobinage 1

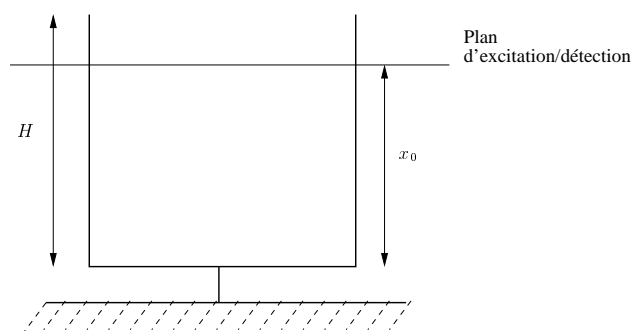


Fig. 11 Plan d'excitation et de détection

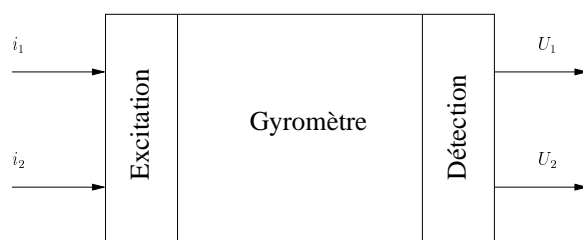


Fig. 12 Structure du gyromètre

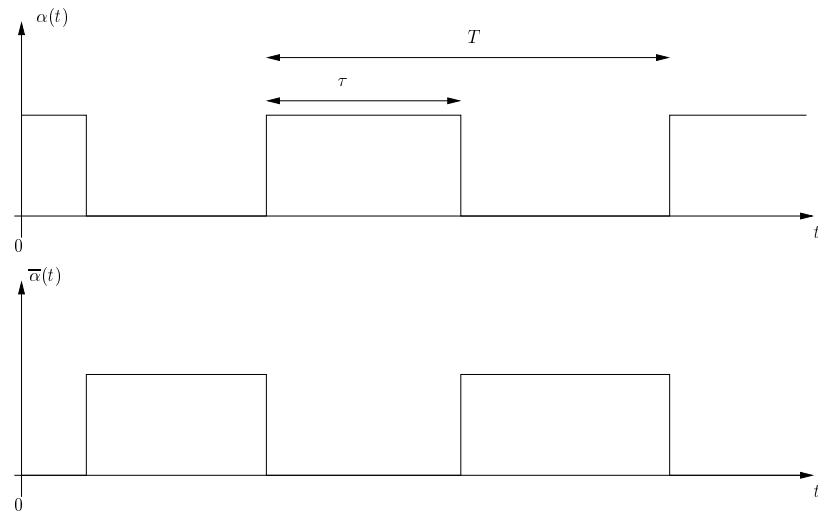
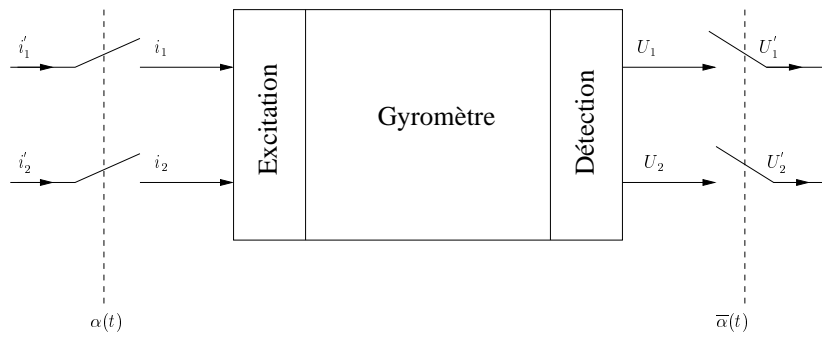
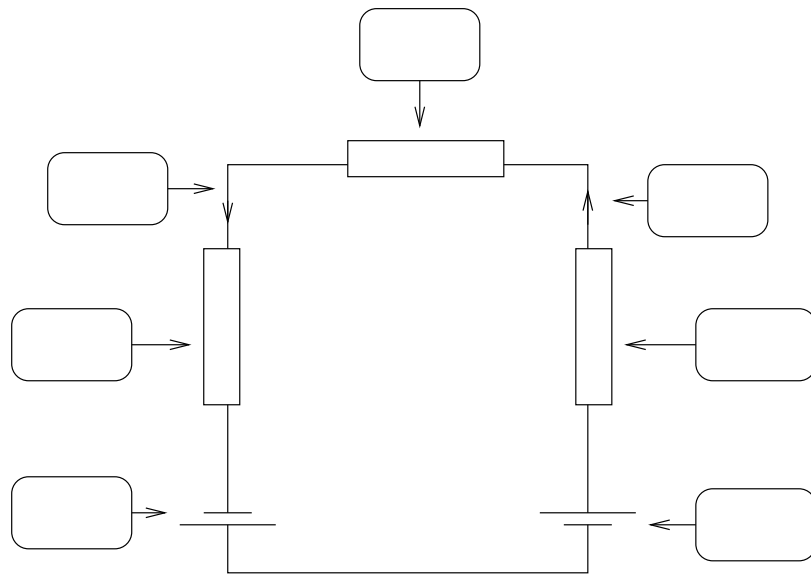


Fig. 13 Gyromètre et séquenceur

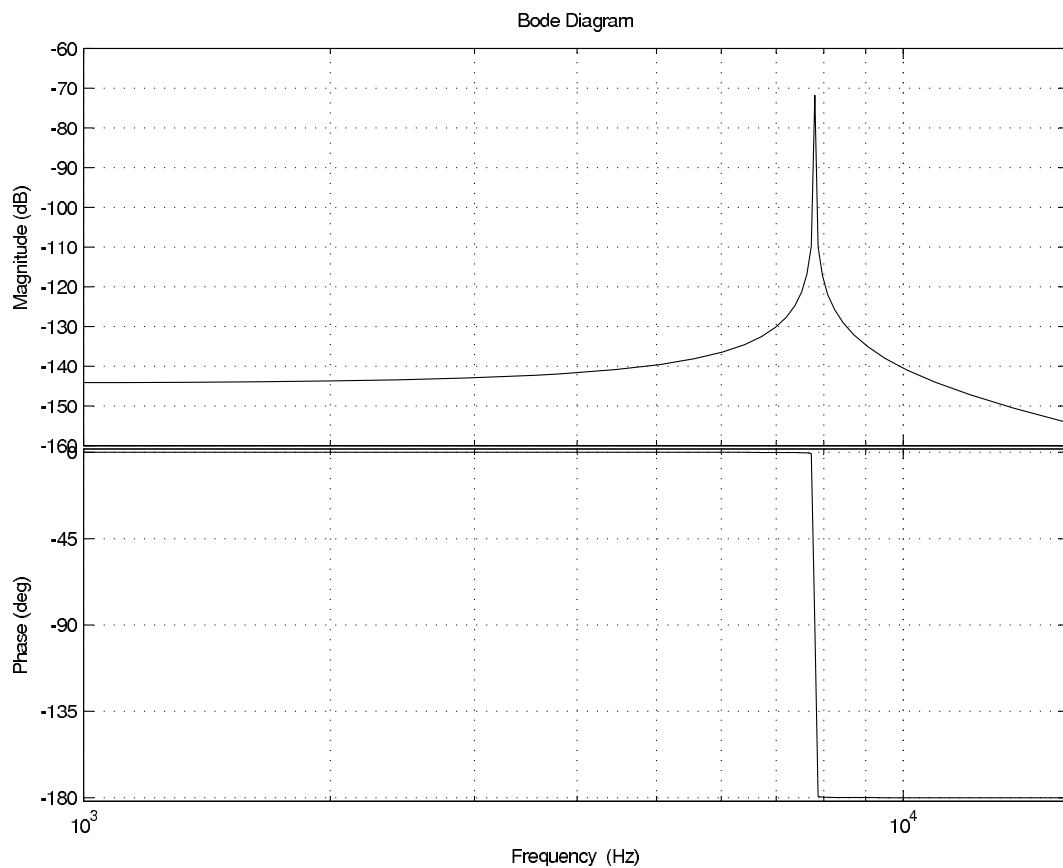
## DOCUMENT REPONSE DE LA PARTIE 1

### Question 1.1. 2



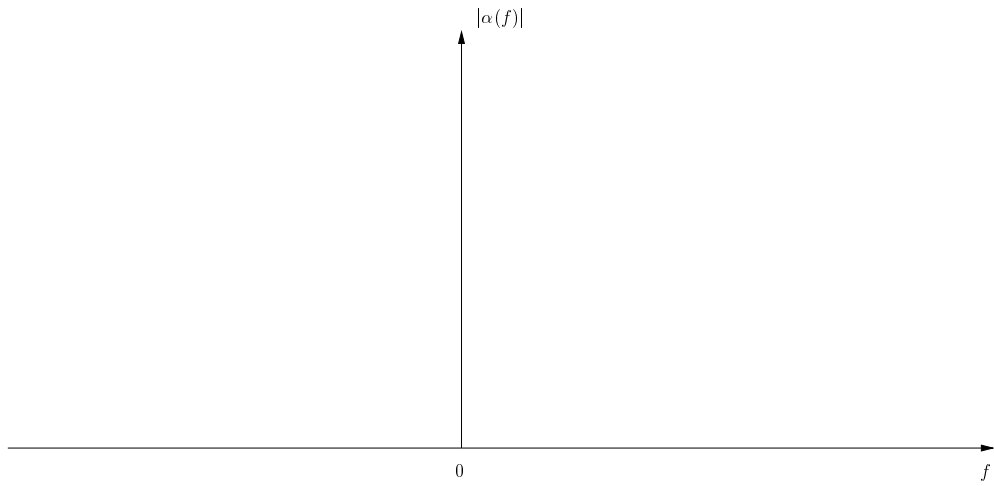
Document réponse 1 Schéma électrique équivalent du circuit magnétique de la figure (7)

### Question 1.3. 4



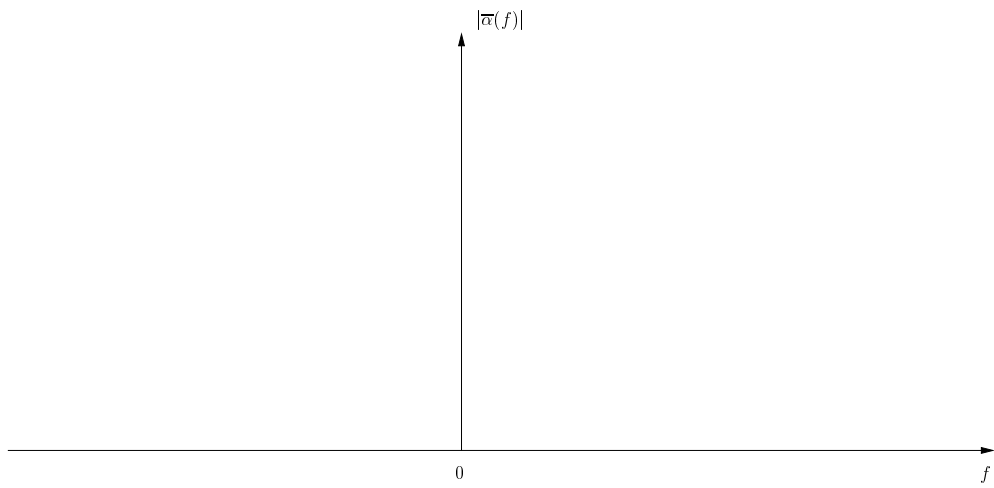
Document réponse 2 Tracé fréquentiel de Bode

**Question 1.5. 3**



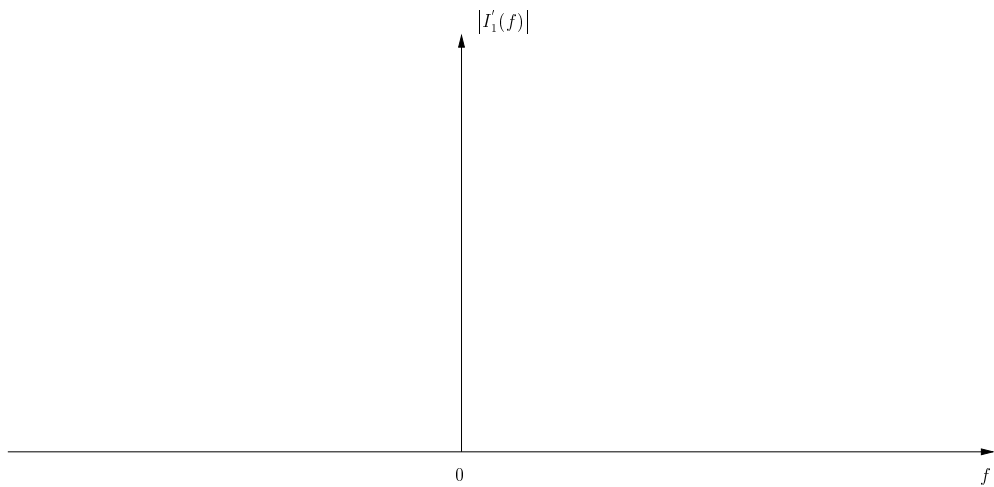
Document réponse 3 Spectre d'amplitude de  $\alpha(t)$

**Question 1.5. 4**



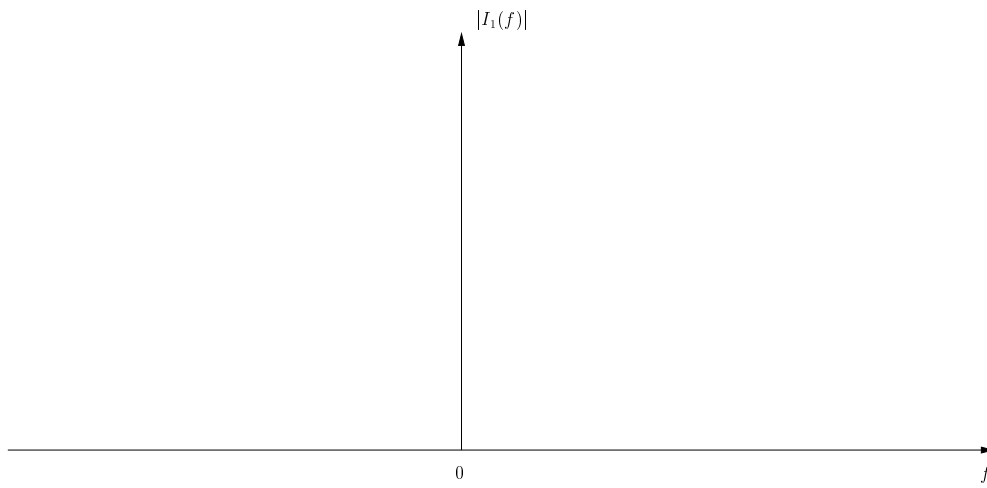
Document réponse 4 Spectre d'amplitude de  $\bar{\alpha}(t)$

**Question 1.5. 8**



Document réponse 5 Spectre d'amplitude de  $i'_1(t)$

**Question 1.5. 9**



Document réponse 6 Spectre d'amplitude de  $i_1(t)$

DOCUMENT REPOSE DE LA PARTIE 3

Question 3.1.11

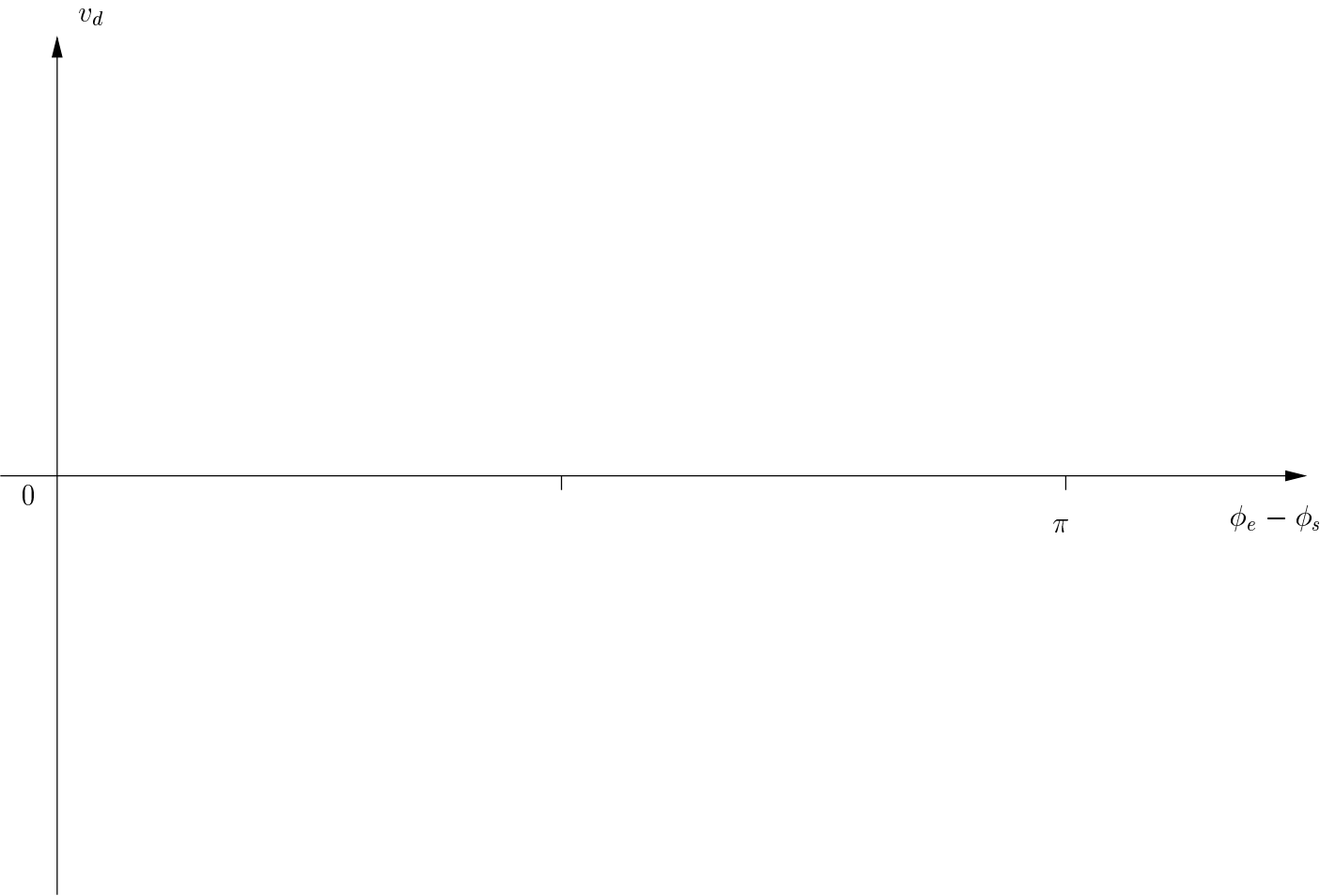
L'angle de rotation est défini par :

$$angle = \sum_{k=0}^i \mu_k . \alpha_k .$$

i	$z_i$	$\alpha_i$	$\mu_i$	$\delta_i$	$x_i$	$y_i$	angle
0	53		+1		1	0	
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

TAB. 1 – Document réponse 7 Calcul du cosinus d'un angle de 53°

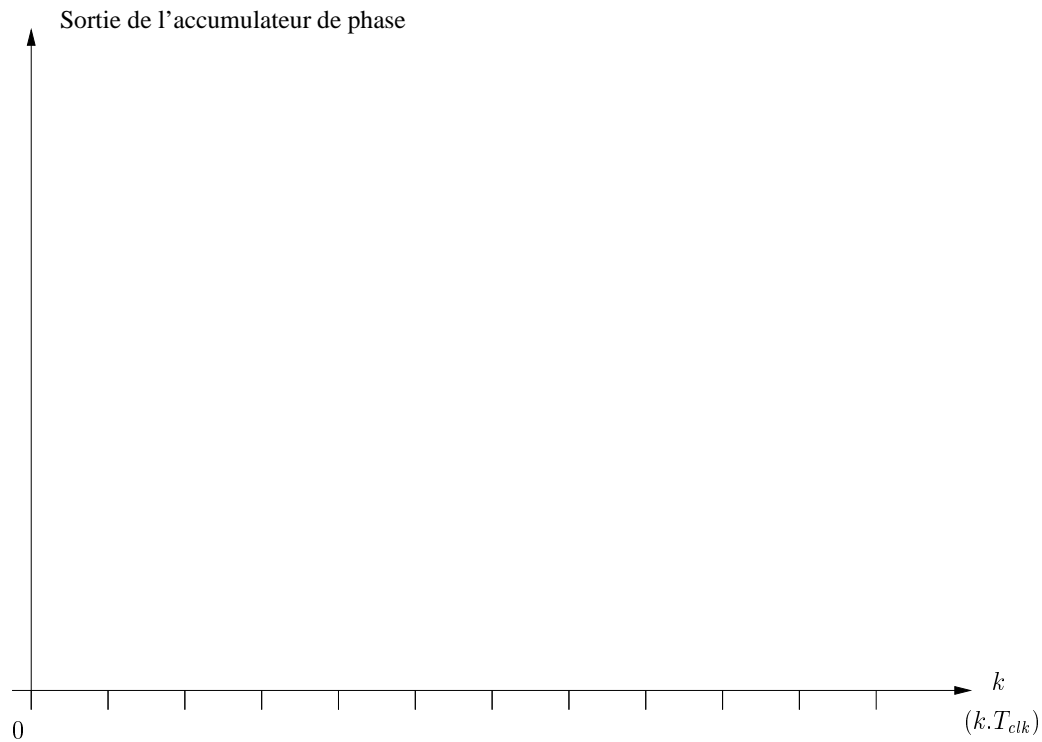
Question 3.2.1.5



Document réponse 8 Caractéristique de comparateur de phase

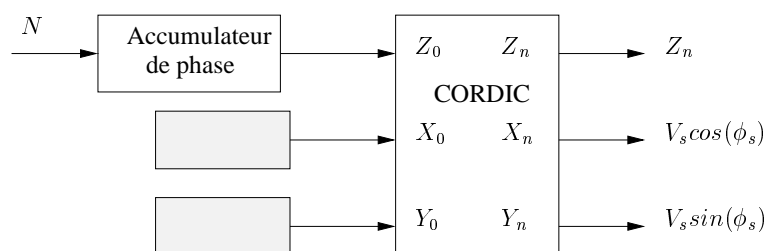


### Question 3.2.2.2



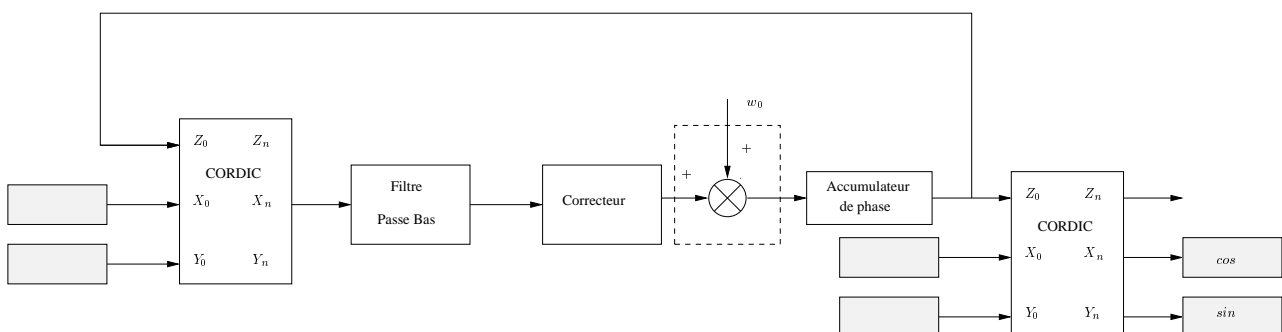
Document réponse 9 Réponse de l'accumulateur de phase numérique

### Question 3.2.2.6



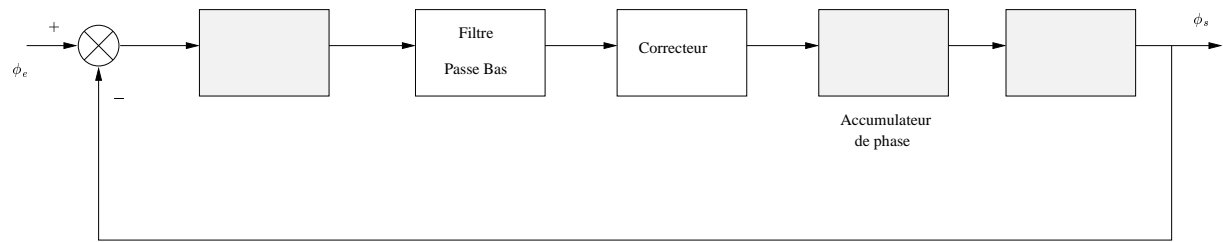
Document réponse 10 Oscillateur numérique

### Question 3.2.2.8



Document réponse 11 PLL entièrement numérique

### Question 3.2.2.9



Document réponse 12 Modélisation de la PLL



## Spartan-II 2.5V FPGA Family: Functional Description

DS001-2 (v2.2) September 3, 2003

### Product Specification

## Architectural Description

### Spartan-II Array

The Spartan-II user-programmable gate array, shown in **Figure 1**, is composed of five major configurable elements:

- IOBs provide the interface between the package pins and the internal logic
- CLBs provide the functional elements for constructing most logic
- Dedicated block RAM memories of 4096 bits each
- Clock DLLs for clock-distribution delay compensation and clock domain control
- Versatile multi-level interconnect structure

As can be seen in **Figure 1**, the CLBs form the central logic structure with easy access to all support and routing structures. The IOBs are located around all the logic and memory elements for easy and quick routing of signals on and off the chip.

Values stored in static memory cells control all the configurable logic elements and interconnect resources. These values load into the memory cells on power-up, and can reload if necessary to change the function of the device.

Each of these elements will be discussed in detail in the following sections.

### Input/Output Block

The Spartan-II IOB, as seen in **Figure 1**, features inputs and outputs that support a wide variety of I/O signaling standards. These high-speed inputs and outputs are capable of supporting various state of the art memory and bus interfaces. **Table 1** lists several of the standards which are supported along with the required reference, output and termination voltages needed to meet the standard.

The three IOB registers function either as edge-triggered D-type flip-flops or as level-sensitive latches. Each IOB has a clock signal (CLK) shared by the three registers and independent Clock Enable (CE) signals for each register.

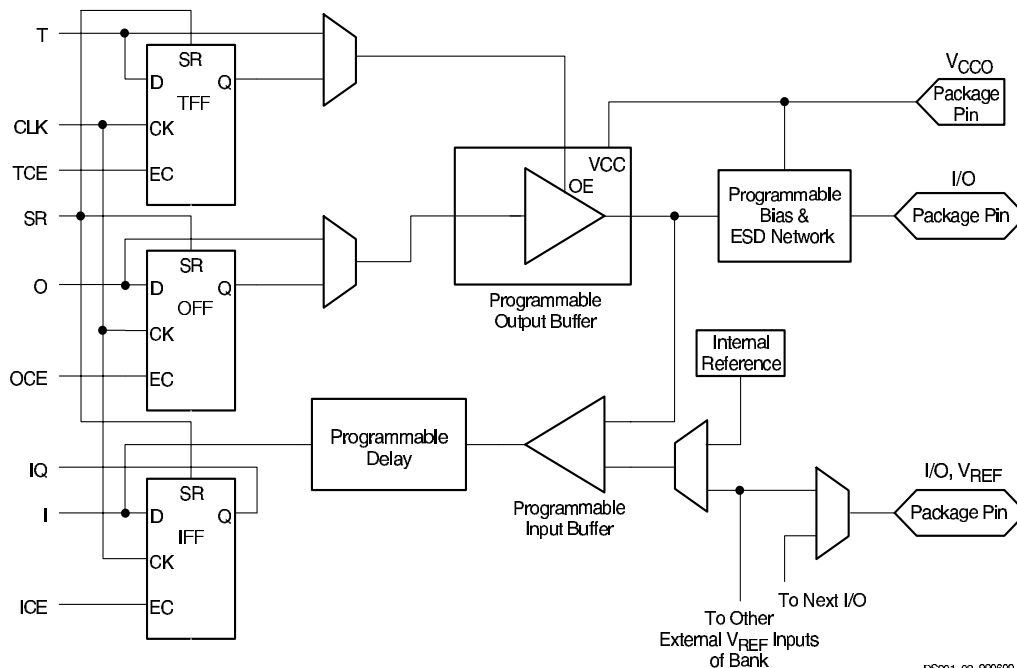


Figure 1: Spartan-II Input/Output Block (IOB)

DS001\_02\_090600

© 2003 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

DS001-2 (v2.2) September 3, 2003  
Product Specification

[www.xilinx.com](http://www.xilinx.com)  
1-800-255-7778

Module 2 of 4  
1

In addition to the CLK and CE control signals, the three registers share a Set/Reset (SR). For each register, this signal can be independently configured as a synchronous Set, a synchronous Reset, an asynchronous Preset, or an asynchronous Clear.

A feature not shown in the block diagram, but controlled by the software, is polarity control. The input and output buffers and all of the IOB control signals have independent polarity controls.

Optional pull-up and pull-down resistors and an optional weak-keeper circuit are attached to each pad. Prior to configuration all outputs not involved in configuration are forced into their high-impedance state. The pull-down resistors and the weak-keeper circuits are inactive, but inputs may optionally be pulled up.

Table 1: Standards Supported by I/O (Typical Values)

I/O Standard	Input Reference Voltage ( $V_{REF}$ )	Output Source Voltage ( $V_{CCO}$ )	Board Termination Voltage ( $V_{TT}$ )
LVTTL (2-24 mA)	N/A	3.3	N/A
LVC MOS2	N/A	2.5	N/A
PCI (3V/5V, 33 MHz/66 MHz)	N/A	3.3	N/A
GTL	0.8	N/A	1.2
GTL+	1.0	N/A	1.5
HSTL Class I	0.75	1.5	0.75
HSTL Class III	0.9	1.5	1.5
HSTL Class IV	0.9	1.5	1.5
SSTL3 Class I and II	1.5	3.3	1.5
SSTL2 Class I and II	1.25	2.5	1.25
CTT	1.5	3.3	1.5
AGP-2X	1.32	3.3	N/A

The activation of pull-up resistors prior to configuration is controlled on a global basis by the configuration mode pins. If the pull-up resistors are not activated, all the pins will float. Consequently, external pull-up or pull-down resistors must be provided on pins required to be at a well-defined logic level prior to configuration.

All pads are protected against damage from electrostatic discharge (ESD) and from over-voltage transients. Two forms of over-voltage protection are provided, one that permits 5V compliance, and one that does not. For 5V compliance, a zener-like structure connected to ground turns on when the output rises to approximately 6.5V. When 5V com-

pliance is not required, a conventional clamp diode may be connected to the output supply voltage,  $V_{CCO}$ . The type of over-voltage protection can be selected independently for each pad.

All Spartan-II IOBs support IEEE 1149.1-compatible boundary scan testing.

### Input Path

A buffer in the Spartan-II IOB input path routes the input signal either directly to internal logic or through an optional input flip-flop.

An optional delay element at the D-input of this flip-flop eliminates pad-to-pad hold time. The delay is matched to the internal clock-distribution delay of the FPGA, and when used, assures that the pad-to-pad hold time is zero.

Each input buffer can be configured to conform to any of the low-voltage signaling standards supported. In some of these standards the input buffer utilizes a user-supplied threshold voltage,  $V_{REF}$ . The need to supply  $V_{REF}$  imposes constraints on which standards can be used in close proximity to each other. See [I/O Banking, page 3](#).

There are optional pull-up and pull-down resistors at each input for use after configuration.

### Output Path

The output path includes a 3-state output buffer that drives the output signal onto the pad. The output signal can be routed to the buffer directly from the internal logic or through an optional IOB output flip-flop.

The 3-state control of the output can also be routed directly from the internal logic or through a flip-flop that provides synchronous enable and disable.

Each output driver can be individually programmed for a wide range of low-voltage signaling standards. Each output buffer can source up to 24 mA and sink up to 48 mA. Drive strength and slew rate controls minimize bus transients.

In most signaling standards, the output high voltage depends on an externally supplied  $V_{CCO}$  voltage. The need to supply  $V_{CCO}$  imposes constraints on which standards can be used in close proximity to each other. See [I/O Banking](#).

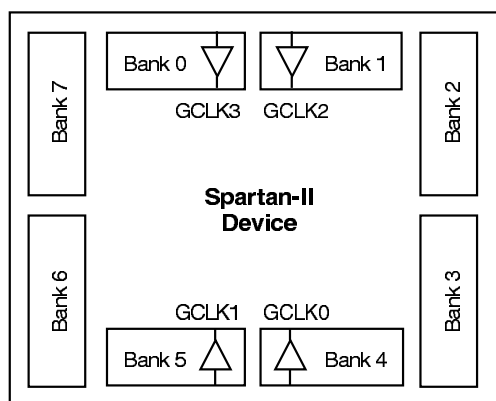
An optional weak-keeper circuit is connected to each output. When selected, the circuit monitors the voltage on the pad and weakly drives the pin High or Low to match the input signal. If the pin is connected to a multiple-source signal, the weak keeper holds the signal in its last state if all drivers are disabled. Maintaining a valid logic level in this way helps eliminate bus chatter.

Because the weak-keeper circuit uses the IOB input buffer to monitor the input level, an appropriate  $V_{REF}$  voltage must be provided if the signaling standard requires one. The provision of this voltage must comply with the I/O banking rules.

## I/O Banking

Some of the I/O standards described above require  $V_{CCO}$  and/or  $V_{REF}$  voltages. These voltages are externally connected to device pins that serve groups of IOBs, called banks. Consequently, restrictions exist about which I/O standards can be combined within a given bank.

Eight I/O banks result from separating each edge of the FPGA into two banks (see Figure 2). Each bank has multiple  $V_{CCO}$  pins which must be connected to the same voltage. Voltage is determined by the output standards in use.



DS001\_03\_060100

Figure 2: Spartan-II I/O Banks

Within a bank, output standards may be mixed only if they use the same  $V_{CCO}$ . Compatible standards are shown in Table 2. GTL and GTL+ appear under all voltages because their open-drain outputs do not depend on  $V_{CCO}$ .

Table 2: Compatible Output Standards

$V_{CCO}$	Compatible Standards
3.3V	PCI, LVTTTL, SSTL3 I, SSTL3 II, CTT, AGP, GTL, GTL+
2.5V	SSTL2 I, SSTL2 II, LVCMOS2, GTL, GTL+
1.5V	HSTL I, HSTL III, HSTL IV, GTL, GTL+

Some input standards require a user-supplied threshold voltage,  $V_{REF}$ . In this case, certain user-I/O pins are automatically configured as inputs for the  $V_{REF}$  voltage. About one in six of the I/O pins in the bank assume this role.

$V_{REF}$  pins within a bank are interconnected internally and consequently only one  $V_{REF}$  voltage can be used within

each bank. All  $V_{REF}$  pins in the bank, however, must be connected to the external voltage source for correct operation.

In a bank, inputs requiring  $V_{REF}$  can be mixed with those that do not but only one  $V_{REF}$  voltage may be used within a bank. Input buffers that use  $V_{REF}$  are not 5V tolerant. LVTTTL, LVCMOS2, and PCI are 5V tolerant. The  $V_{CCO}$  and  $V_{REF}$  pins for each bank appear in the device pinout tables.

Within a given package, the number of  $V_{REF}$  and  $V_{CCO}$  pins can vary depending on the size of device. In larger devices, more I/O pins convert to  $V_{REF}$  pins. Since these are always a superset of the  $V_{REF}$  pins used for smaller devices, it is possible to design a PCB that permits migration to a larger device. All  $V_{REF}$  pins for the largest device anticipated must be connected to the  $V_{REF}$  voltage, and not used for I/O.

Table 3: Independent Banks Available

Package	VQ100 PQ208	CS144 TQ144	FG256 FG456
Independent Banks	1	4	8

## Configurable Logic Block

The basic building block of the Spartan-II CLB is the logic cell (LC). An LC includes a 4-input function generator, carry logic, and storage element. Output from the function generator in each LC drives the CLB output and the D input of the flip-flop. Each Spartan-II CLB contains four LCs, organized in two similar slices; a single slice is shown in Figure 3.

In addition to the four basic LCs, the Spartan-II CLB contains logic that combines function generators to provide functions of five or six inputs.

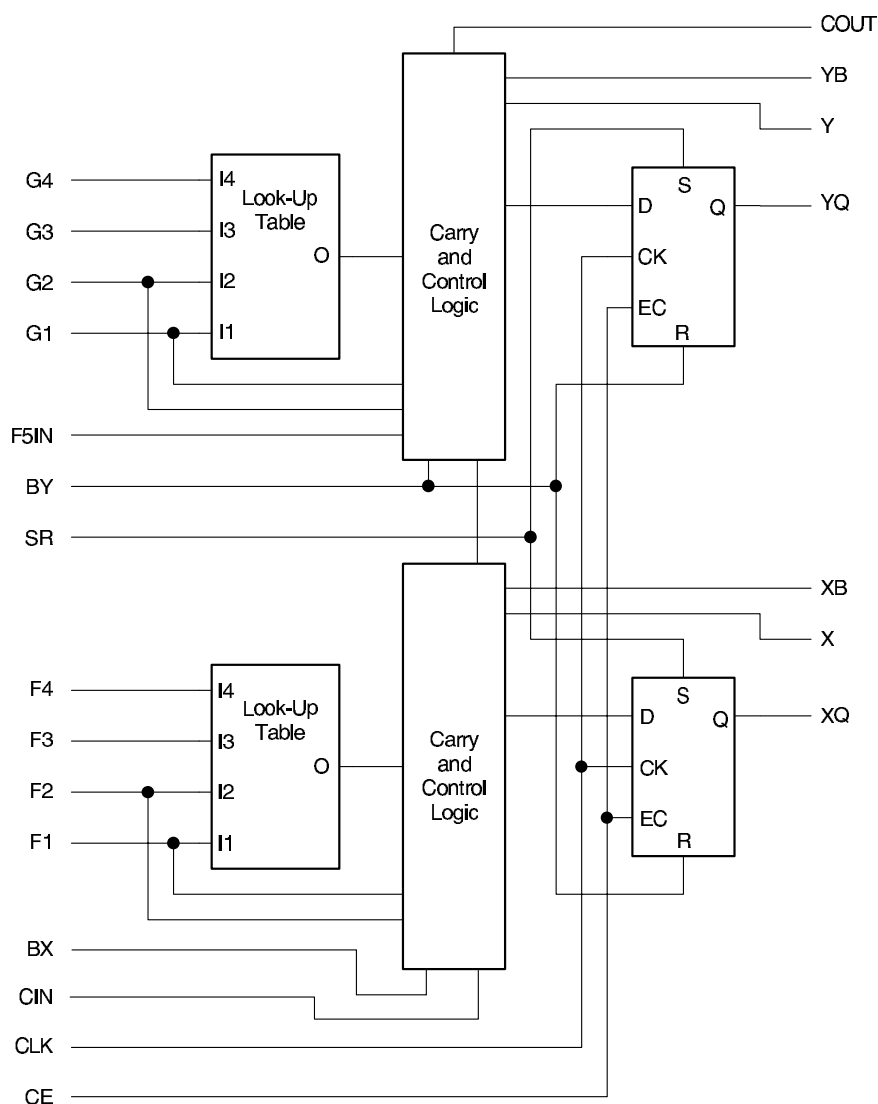
## Look-Up Tables

Spartan-II function generators are implemented as 4-input look-up tables (LUTs). In addition to operating as a function generator, each LUT can provide a 16 x 1-bit synchronous RAM. Furthermore, the two LUTs within a slice can be combined to create a 16 x 2-bit or 32 x 1-bit synchronous RAM, or a 16 x 1-bit dual-port synchronous RAM.

The Spartan-II LUT can also provide a 16-bit shift register that is ideal for capturing high-speed or burst-mode data. This mode can also be used to store data in applications such as Digital Signal Processing.

## Storage Elements

Storage elements in the Spartan-II slice can be configured either as edge-triggered D-type flip-flops or as level-sensitive latches. The D inputs can be driven either by function generators within the slice or directly from slice inputs, bypassing the function generators.



DS001\_04\_091400

Figure 3: **Spartan-II CLB Slice** (two identical slices in each CLB)

In addition to Clock and Clock Enable signals, each slice has synchronous set and reset signals (SR and BY). SR forces a storage element into the initialization state specified for it in the configuration. BY forces it into the opposite state. Alternatively, these signals may be configured to operate asynchronously.

All control signals are independently invertible, and are shared by the two flip-flops within the slice.

### Additional Logic

The F5 multiplexer in each slice combines the function generator outputs. This combination provides either a function generator that can implement any 5-input function, a 4:1 multiplexer, or selected functions of up to nine inputs.

Similarly, the F6 multiplexer combines the outputs of all four function generators in the CLB by selecting one of the F5-multiplexer outputs. This permits the implementation of any 6-input function, an 8:1 multiplexer, or selected functions of up to 19 inputs.

Each CLB has four direct feedthrough paths, one per LC. These paths provide extra data input lines or additional local routing that does not consume logic resources.

### Arithmetic Logic

Dedicated carry logic provides fast arithmetic carry capability for high-speed arithmetic functions. The Spartan-II CLB supports two separate carry chains, one per slice. The height of the carry chains is two bits per CLB.

The arithmetic logic includes an XOR gate that allows a 1-bit full adder to be implemented within an LC. In addition, a dedicated AND gate improves the efficiency of multiplier implementation.

The dedicated carry path can also be used to cascade function generators for implementing wide logic functions.

### BUFTs

Each Spartan-II CLB contains two 3-state drivers (BUFTs) that can drive on-chip busses. See **Dedicated Routing**, page 6. Each Spartan-II BUFT has an independent 3-state control pin and an independent input pin.

### Block RAM

Spartan-II FPGAs incorporate several large block RAM memories. These complement the distributed RAM Look-Up Tables (LUTs) that provide shallow memory structures implemented in CLBs.

Block RAM memory blocks are organized in columns. All Spartan-II devices contain two such columns, one along each vertical edge. These columns extend the full height of the chip. Each memory block is four CLBs high, and consequently, a Spartan-II device eight CLBs high will contain two memory blocks per column, and a total of four blocks.

Table 4: Spartan-II Block RAM Amounts

Spartan-II Device	# of Blocks	Total Block RAM Bits
XC2S15	4	16K
XC2S30	6	24K
XC2S50	8	32K
XC2S100	10	40K
XC2S150	12	48K
XC2S200	14	56K

Each block RAM cell, as illustrated in Figure 4, is a fully synchronous dual-ported 4096-bit RAM with independent control signals for each port. The data widths of the two ports can be configured independently, providing built-in bus-width conversion.

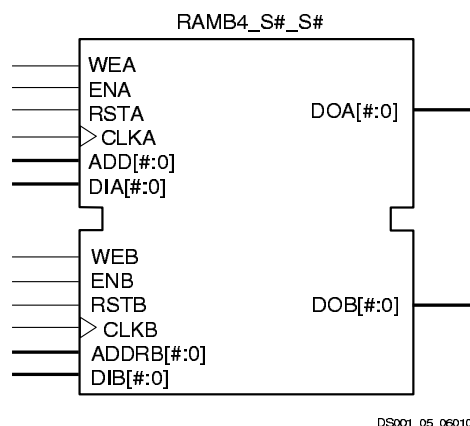


Figure 4: Dual-Port Block RAM

Table 5 shows the depth and width aspect ratios for the block RAM.

Table 5: Block RAM Port Aspect Ratios

Width	Depth	ADDR Bus	Data Bus
1	4096	ADDR<11:0>	DATA<0>
2	2048	ADDR<10:0>	DATA<1:0>
4	1024	ADDR<9:0>	DATA<3:0>
8	512	ADDR<8:0>	DATA<7:0>
16	256	ADDR<7:0>	DATA<15:0>

The Spartan-II block RAM also includes dedicated routing to provide an efficient interface with both CLBs and other block RAMs.

### Programmable Routing Matrix

It is the longest delay path that limits the speed of any worst-case design. Consequently, the Spartan-II routing architecture and its place-and-route software were defined in a single optimization process. This joint optimization minimizes long-path delays, and consequently, yields the best system performance.

The joint optimization also reduces design compilation times because the architecture is software-friendly. Design cycles are correspondingly reduced due to shorter design iteration times.

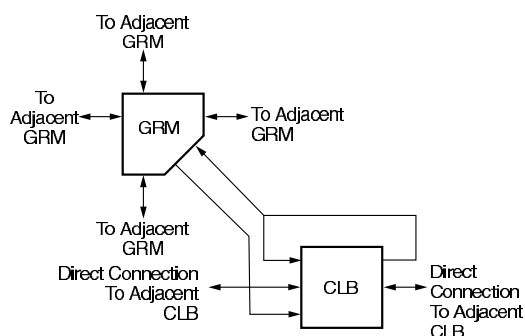
### Local Routing

The local routing resources, as shown in Figure 5, provide the following three types of connections:

- Interconnections among the LUTs, flip-flops, and General Routing Matrix (GRM)
- Internal CLB feedback paths that provide high-speed connections to LUTs within the same CLB, chaining

them together with minimal routing delay

- Direct paths that provide high-speed connections between horizontally adjacent CLBs, eliminating the delay of the GRM



DS001\_06\_032300

Figure 5: **Spartan-II Local Routing**

### General Purpose Routing

Most Spartan-II signals are routed on the general purpose routing, and consequently, the majority of interconnect resources are associated with this level of the routing hierarchy. The general routing resources are located in horizontal and vertical routing channels associated with the rows and columns CLBs. The general-purpose routing resources are listed below.

- Adjacent to each CLB is a General Routing Matrix (GRM). The GRM is the switch matrix through which horizontal and vertical routing resources connect, and is also the means by which the CLB gains access to the general purpose routing.
- 24 single-length lines route GRM signals to adjacent

GRMs in each of the four directions.

- 96 buffered Hex lines route GRM signals to other GRMs six blocks away in each one of the four directions. Organized in a staggered pattern, Hex lines may be driven only at their endpoints. Hex-line signals can be accessed either at the endpoints or at the midpoint (three blocks from the source). One third of the Hex lines are bidirectional, while the remaining ones are unidirectional.
- 12 Longlines are buffered, bidirectional wires that distribute signals across the device quickly and efficiently. Vertical Longlines span the full height of the device, and horizontal ones span the full width of the device.

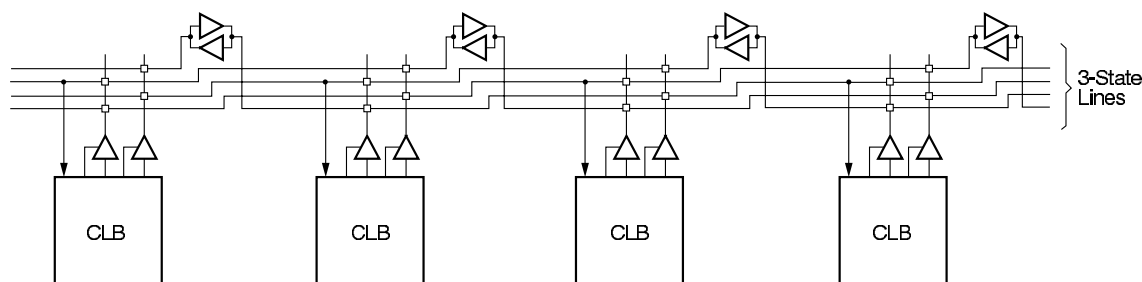
### I/O Routing

Spartan-II devices have additional routing resources around their periphery that form an interface between the CLB array and the IOBs. This additional routing, called the VersaRing, facilitates pin-swapping and pin-locking, such that logic redesigns can adapt to existing PCB layouts. Time-to-market is reduced, since PCBs and other system components can be manufactured while the logic design is still in progress.

### Dedicated Routing

Some classes of signal require dedicated routing resources to maximize performance. In the Spartan-II architecture, dedicated routing resources are provided for two classes of signal.

- Horizontal routing resources are provided for on-chip 3-state busses. Four partitionable bus lines are provided per CLB row, permitting multiple busses within a row, as shown in Figure 6.
- Two dedicated nets per CLB propagate carry signals vertically to the adjacent CLB.



DS001\_07\_080600

Figure 6: **BUFT Connections to Dedicated Horizontal Bus Lines**



## Global Routing

Global Routing resources distribute clocks and other signals with very high fanout throughout the device. Spartan-II devices include two tiers of global routing resources referred to as primary and secondary global routing resources.

- The primary global routing resources are four dedicated global nets with dedicated input pins that are designed to distribute high-fanout clock signals with minimal skew. Each global clock net can drive all CLB, IOB, and block RAM clock pins. The primary global nets may only be driven by global buffers. There are four global buffers, one for each global net.
- The secondary global routing resources consist of 24 backbone lines, 12 across the top of the chip and 12 across bottom. From these lines, up to 12 unique signals per column can be distributed via the 12 longlines in the column. These secondary resources are more flexible than the primary resources since they are not restricted to routing only to clock pins.

## Clock Distribution

The Spartan-II family provides high-speed, low-skew clock distribution through the primary global routing resources described above. A typical clock distribution net is shown in Figure 7.

Four global buffers are provided, two at the top center of the device and two at the bottom center. These drive the four primary global nets that in turn drive any clock pin.

Four dedicated clock pads are provided, one adjacent to each of the global buffers. The input to the global buffer is selected either from these pads or from signals in the general purpose routing. Global clock pins do not have the option for internal, weak pull-up resistors.

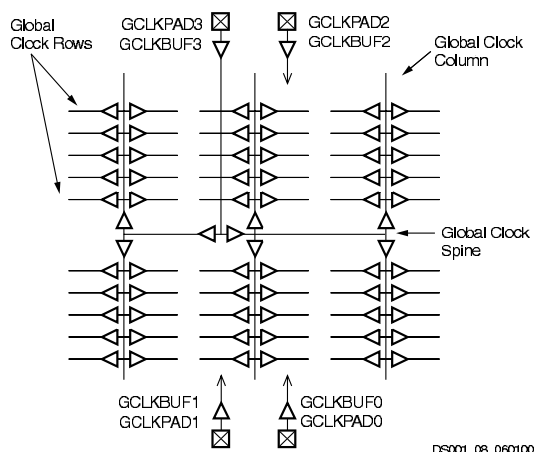


Figure 7: Global Clock Distribution Network

## Delay-Locked Loop (DLL)

Associated with each global clock input buffer is a fully digital Delay-Locked Loop (DLL) that can eliminate skew between the clock input pad and internal clock-input pins throughout the device. Each DLL can drive two global clock networks. The DLL monitors the input clock and the distributed clock, and automatically adjusts a clock delay element. Additional delay is introduced such that clock edges reach internal flip-flops exactly one clock period after they arrive at the input. This closed-loop system effectively eliminates clock-distribution delay by ensuring that clock edges arrive at internal flip-flops in synchronism with clock edges arriving at the input.

In addition to eliminating clock-distribution delay, the DLL provides advanced control of multiple clock domains. The DLL provides four quadrature phases of the source clock, can double the clock, or divide the clock by 1.5, 2, 2.5, 3, 4, 5, 8, or 16. It has six outputs.

The DLL also operates as a clock mirror. By driving the output from a DLL off-chip and then back on again, the DLL can be used to deskew a board level clock among multiple Spartan-II devices.

In order to guarantee that the system clock is operating correctly prior to the FPGA starting up after configuration, the DLL can delay the completion of the configuration process until after it has achieved lock.

## Boundary Scan

Spartan-II devices support all the mandatory boundary-scan instructions specified in the IEEE standard 1149.1. A Test Access Port (TAP) and registers are provided that implement the EXTEST, SAMPLE/PRELOAD, and BYPASS instructions. The TAP also supports two USERCODE instructions and internal scan chains.

The TAP uses dedicated package pins that always operate using LVTTTL. For TDO to operate using LVTTTL, the  $V_{CCO}$  for Bank 2 must be 3.3V. Otherwise, TDO switches rail-to-rail between ground and  $V_{CCO}$ .

Boundary-scan operation is independent of individual IOB configurations, and unaffected by package type. All IOBs, including unbonded ones, are treated as independent 3-state bidirectional pins in a single scan chain. Retention of the bidirectional test capability after configuration facilitates the testing of external interconnections.

Table 6 lists the boundary-scan instructions supported in Spartan-II FPGAs. Internal signals can be captured during EXTEST by connecting them to unbonded or unused IOBs. They may also be connected to the unused outputs of IOBs defined as unidirectional input pins.

Table 6: Boundary-Scan Instructions

Boundary-Scan Command	Binary Code[4:0]	Description
EXTEST	00000	Enables boundary-scan EXTEST operation
SAMPLE	00001	Enables boundary-scan SAMPLE operation
USR1	00010	Access user-defined register 1
USR2	00011	Access user-defined register 2
CFG_OUT	00100	Access the configuration bus for Readback
CFG_IN	00101	Access the configuration bus for Configuration
INTEST	00111	Enables boundary-scan INTEST operation
USRCODE	01000	Enables shifting out USER code
IDCODE	01001	Enables shifting out of ID Code
HIZ	01010	Disables output pins while enabling the Bypass Register
JSTART	01100	Clock the start-up sequence when StartupClk is TCK
BYPASS	11111	Enables BYPASS
RESERVED	All other codes	Xilinx reserved instructions

The public boundary-scan instructions are available prior to configuration. After configuration, the public instructions remain available together with any USERCODE instructions installed during the configuration. While the SAMPLE and BYPASS instructions are available during configuration, it is recommended that boundary-scan operations not be performed during this transitional period.

In addition to the test instructions outlined above, the boundary-scan circuitry can be used to configure the FPGA, and also to read back the configuration data.

To facilitate internal scan chains, the User Register provides three outputs (Reset, Update, and Shift) that represent the corresponding states in the boundary-scan internal state machine.

Figure 8 is a diagram of the Spartan-II family boundary scan logic. It includes three bits of Data Register per IOB, the IEEE 1149.1 Test Access Port controller, and the Instruction Register with decodes.

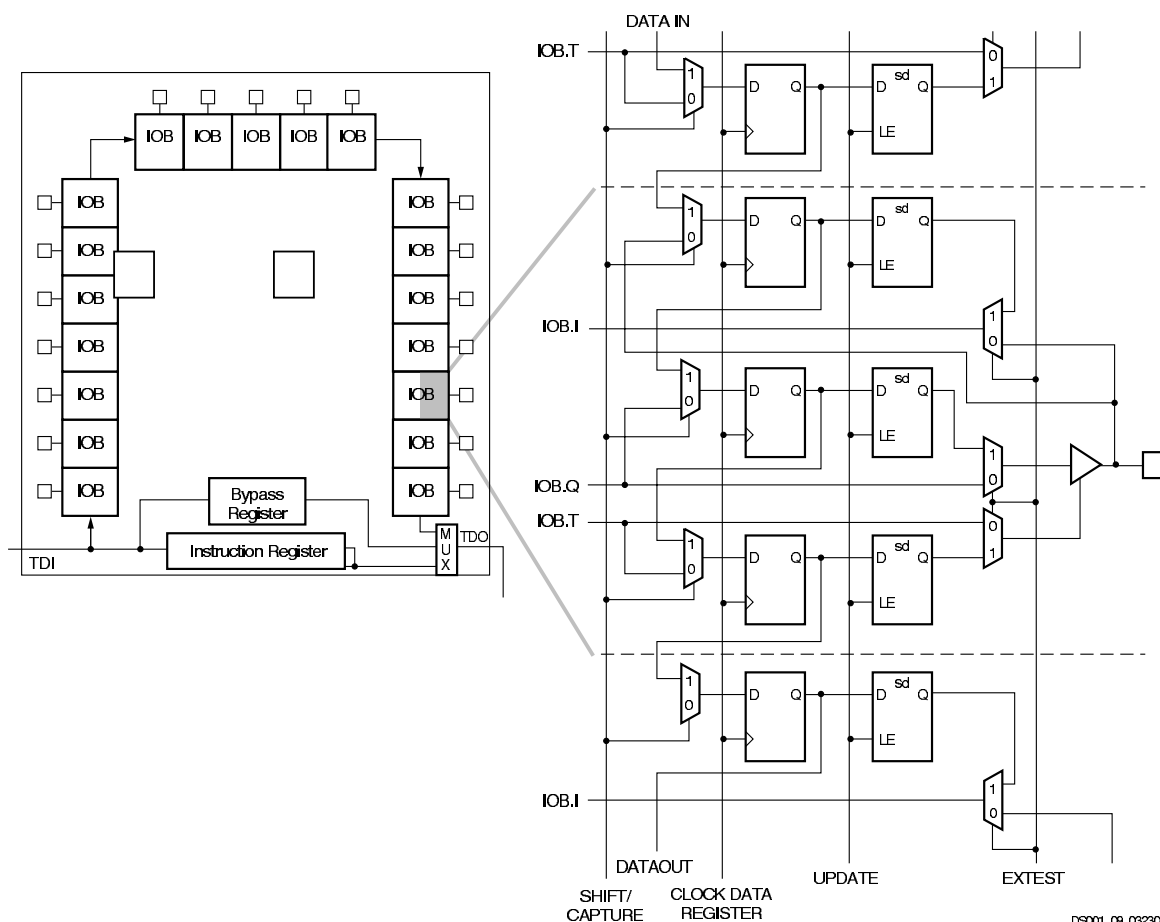


Figure 8: Spartan-II Family Boundary Scan Logic

### Bit Sequence

The bit sequence within each IOB is: In, Out, 3-State. The input-only pins contribute only the In bit to the boundary scan I/O data register, while the output-only pins contribute all three bits.

From a cavity-up view of the chip (as shown in the FPGA Editor), starting in the upper right chip corner, the boundary scan data-register bits are ordered as shown in Figure 9.

BSDL (Boundary Scan Description Language) files for Spartan-II family devices are available on the Xilinx website, in the File Download area.