Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 71 (1980)

Heft: 3

Artikel: Méthode de synthèse des systèmes digitaux : première partie : l'unité de traitement

Autor: Stauffer, A.

DOI: https://doi.org/10.5169/seals-905224

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 10.12.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Méthode de synthèse des systèmes digitaux
Première partie: l'unité de traitement

Par A. Stauffer

681.32

Tout système digital se compose d'une unité de traitement et d'une unité de commande. La méthode de synthèse de l'unité de traitement d'un tel système repose sur l'emploi d'un langage de description. Appliqué aux composants combinatoires et séquentiels, c'est-à-dire aux circuits intégrés du marché qui ne traitent pas les bits de manière individuelle, mais opèrent sur des ensembles de bits, ce langage conduit à une réalisation câblée de l'unité considérée. La mise en œuvre de la méthode est illustrée par la synthèse de l'unité de traitement d'un système digital réalisant la division de deux nombres binaires.

Jedes digitale System besteht aus einem Prozessor und einer Steuereinheit. Zur Synthese des Prozessors wird eine beschreibende Sprache verwendet. Diese Sprache führt zu einer verdrahteten Lösung, wenn man sie auf verknüpfende und sequentielle Bauteile anwendet, d.h. auf marktgängige IC, welche nicht einzelne Bit sondern Gruppen von Bit, also Worte, verarbeiten. Die Methode wird am Beispiel der Synthese des Prozessors eines digitalen Systems veranschaulicht, das zwei binäre Zahlen dividiert.

1. Introduction

La méthode de synthèse de l'unité de traitement de données et de l'unité de commande d'un système digital repose sur l'emploi d'un langage de description. Elle conduit à une réalisation câblée de l'unité de traitement et à une réalisation programmée de l'unité de commande.

La réalisation câblée de l'unité de traitement résulte de l'interconnexion de composants combinatoires et séquentiels disponibles sur le marché sous forme de circuits intégrés à moyenne échelle. Ces composants ne traitent pas les bits de manière individuelle, mais opèrent sur des ensembles de bits: les *mots*. Il s'agit essentiellement de multiplexeurs et d'opérateurs arithmétiques ou logiques, pour les composants combinatoires, de compteurs et de registres à décalage, pour les composants séquentiels. Selon leur nature combinatoire ou séquentielle, ces composants et leurs assemblages constituent respectivement les *ressources de calcul* et les *registres de transfert* de l'unité de traitement.

Le présent article définit un langage de description original et le met en œuvre pour décrire un certain nombre de composants combinatoires et séquentiels. Il applique ensuite ce langage aux différentes étapes de la synthèse de l'unité de traitement d'un système digital réalisant la division de deux nombres binaires. La synthèse de l'unité de commande du système digital considéré fera l'objet d'un second article[1]).

2. Langage de description

2.1 Généralités

Il existe un grand nombre de langages de description destinés à faciliter la synthèse des systèmes digitaux [1. pp. 107 à 146; 2]. Le langage défini ci-après s'en inspire. Son originalité résulte de l'emploi d'un signe *égal* (=), en plus de la flèche (←) propre aux transferts d'information. L'usage de ces deux signes permet de décrire à la fois les opérations combinatoires effectuées par les ressources de calcul et les opérations séquentielles réalisées par les registres de transfert.

Dans ce langage, une ressource est en général désignée par des lettres minuscules et un registre ou une mémoire par des lettres majuscules. La numérotation des bits s'effectue de droite à gauche, en partant de zéro (fig. 1a).

[1]) La seconde partie, consacrée à l'unité de commande, sera publiée dans le Bulletin ASE/UCS 71(1980)7, du 12 avril 1980.

2.2 Déclarations

Les déclarations permettent d'indiquer les dimensions et de repérer les bits d'une ressource, d'un registre ou d'une mémoire, par exemple

a < 1×4 >:	ressource a à 4 bits;
b < 0 >:	bit 0 de la ressource b;
c < 7,5 >:	bits 7 et 5 de la ressource c;
d < 11–8 >:	bits 11 à 8 de la ressource d;
A < 1×8 >:	registre A à 8 bits;
B < 7 >:	bit 7 du registre B;
C < 3,1 >:	bits 3 et 1 du registre C;
D < 7–4 >:	bits 7 à 4 du registre D;
M < 32×8 >:	mémoire M à 32×8 bits.

La position de la mémoire M adressée par le registre A est notée M(A). Ainsi

M(15) < 3 >:	bit 3 de la position d'adresse 15 de la mémoire M;
M(7) < 4,2 >:	bits 4 et 2 de la position d'adresse 7 de la mémoire M;
M(2) < 5 – 3 >:	bits 5 à 3 de la position d'adresse 2 de la mémoire M.

2.3 Opérateurs

Les opérateurs arithmétiques et logiques sont

+ :	somme arithmétique;
– :	différence arithmétique;
$\overline{A}$:	NON logique;
∧ :	ET logique;
∨ :	OU logique;
(+):	OU-exclusif logique.

2.4 Transferts

Les opérations de transfert sont indiquées par des flèches, par exemple

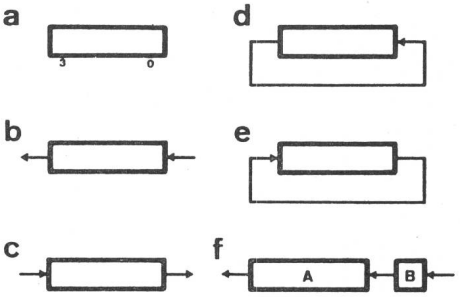


Fig. 1 Ressources ou registres et décalages

$A \leftarrow B$: le contenu du registre B est chargé dans le registre A;
$B \leftarrow M(A)$: le contenu de la position de la mémoire M, adressée par le registre A, est chargé dans le registre B.

Pour des registres à 1 bit, l'opération de transfert $A \leftarrow B$ est équivalente à la notation $A^+ = B$ du modèle synchrone des bascules bistables [3, pp. 87...115].

2.5 *Décalages*

Les opérations de décalage sont

SHL: décalage à gauche (SHift Left) de 1 bit (fig. 1b);
SHR: décalage à droite (SHift Right) de 1 bit (fig. 1c);
ROTL: rotation à gauche de 1 bit (fig. 1d);
ROTR: rotation à droite de 1 bit (fig. 1e).

Pour les registres accolés, les décalages sont du type

SHL A, B: décalage à gauche de 1 bit du contenu des registres accolés A et B (fig. 1f).

2.6 *Incrémentation-décrémentation*

Les opérations d'incrémentation et de décrémentation sont

$A \leftarrow A + 1$: incrémentation du contenu du registre A;
$A \leftarrow A - 1$: décrémentation du contenu du registre A.

2.7 *Fonctions*

Les opérations combinatoires sont indiquées par des signes *égal*, par exemple

$a = A \wedge B$: la fonction combinatoire réalisée par la ressource a correspond au produit logique bit à bit du contenu du registre A par le contenu du registre B.

2.8 *Simultanéités*

Les opérations simultanées sont représentées sur la même ligne, par exemple

$A \leftarrow C$; $B \leftarrow D$: transfert du contenu des registres C et D dans les registres A et B.

2.9 *Opération neutre*

Les opérations du type $A \leftarrow A$ sont toujours implicites lorsque des opérations de type différent s'effectuent simultanément. Une notation résumée permet toutefois de les expliciter quand elles s'appliquent à l'ensemble des registres en l'absence d'opérations d'un autre type:

NOP: opération neutre (NO OPeration).

3. Composants

3.1 *Généralités*

Les composants sont des systèmes logiques combinatoires ou séquentiels qui ne traitent pas les bits de manière individuelle, mais opèrent sur des mots. Il s'agit essentiellement de multiplexeurs, d'opérateurs arithmétiques ou logiques, de compteurs et de registres, disponibles sur le marché sous forme de circuits intégrés à moyenne échelle [4, pp. 83...112].

3.2 *Composant combinatoire*

Un composant combinatoire à n bits est un système logique combinatoire (SLC) à n sorties $Z_1, Z_2, ..., Z_n$; par commodité, on répartit ses $(l + m)$ entrées en l variables de commande $C_1, C_2, ..., C_l$ et m variables d'information $X_1, X_2, ..., X_m$. La figure 2a donne le schéma général d'un tel composant; dans ce schéma, les variables d'information sont représentées par les flèches verticales et les variables de commande par les flèches horizontales.

Compte tenu du nombre de variables relativement élevé du composant combinatoire, la table de vérité de celui-ci ne constitue plus un mode de représentation très adéquat. Il est préférable de lui substituer une *table des opérations* semblable à celle de la figure 2b. Le langage de description utilisé dans cette table permet de définir les opérations combinatoires exécutées par le composant sur ses variables d'information, en fonction de ses variables de commande.

Parmi les composants combinatoires disponibles sous forme de circuits intégrés TTL [5], les plus utilisés sont

l'opérateur ET (7408): ce composant admet deux schémas (fig. 2c et 2e) et deux tables des opérations distinctes (fig. 2d et 2f) selon que les variables 4A, 3A, 2A et 1A sont des variables d'information individuelles ou une variable de commande commune A;

l'additionneur à 4 bits (7483): le schéma et la table des opérations de ce composant apparaissent aux figures 2g et 2h;

l'opérateur OU-exclusif (7486): ce composant admet les schémas et les tables des opérations représentés aux figures 2i, 2j, 2k et 2l;

le multiplexeur à une variable (74157): les figures 2m et 2n décrivent ce composant.

Dans la table des opérations des composants précédents, les ensembles de bits entre parenthèses constituent des mots. Lorsque le composant réalise des opérations arithmétiques, ces mots représentent chacun un nombre binaire.

3.3 *Composant séquentiel*

Un composant séquentiel à n bits est une machine séquentielle synchronisée (MSS) à n sorties $Y_1, Y_2, ..., Y_n$ dont on répartit, par commodité, les $(l + m)$ entrées en l variables de commande $C_1, C_2, ..., C_l$ et m variables d'information $X_1, X_2, ..., X_m$. Cette machine est synchronisée par le signal d'horloge CK. La figure 3a donne le schéma général du composant séquentiel. Les variables de commande et d'information correspondent aux flèches horizontales et verticales.

La table des opérations de la figure 3b constitue un mode de représentation mieux adapté au composant séquentiel que la table d'états traditionnelle. Elle utilise le langage de description pour définir les opérations séquentielles exécutées par le composant sur ses variables d'information en fonction de ses variables de commande.

Parmi les composants séquentiels disponibles sous forme de circuits intégrés TTL [5], les plus courants sont

la bascule JK (7476): le schéma et la table des opérations de ce composant apparaissent aux figures 3c et 3d;

le compteur par 16 (74163): ce composant, dont les figures 3e et 3f fournissent le schéma et la table des opérations, comporte une sortie combinatoire CO égale au produit logique de Q_D, Q_C, Q_B, Q_A et T;

le registre à décalage (74178): ce composant admet deux schémas (fig. 3g et 3i) et deux tables des opérations distinctes (fig. 3h et 3j) selon que l'entrée SI est considérée comme variable d'information ou de commande;

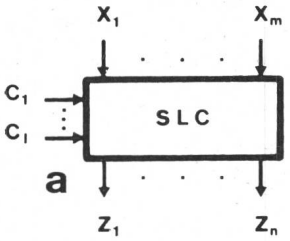
le multiplexeur à une variable avec mémorisation (74298): ce composant apparaît aux figures 3k et 3l.

3.4 *Assemblage combinatoire*

Un assemblage combinatoire à n bits est un système logique combinatoire à n sorties obtenu par l'interconnexion de composants combinatoires à n bits.

L'additionneur-soustracteur à 4 bits (7483 + 7486) de la figure 4a constitue un exemple d'assemblage combinatoire. Il réalise la soustraction d'un nombre par addition du complé-

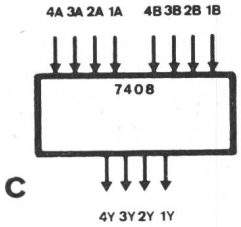
Fig. 2
Schémas et tables des opérations de composants combinatoires



a

opération	description	$C_1 \dots C_\ell$
OP_0	$(Z_1, \dots, Z_n) = f_0 \quad (X_1, \dots, X_m)$	0 ... 0
⋮		
$OP_{2^\ell-1}$	$(Z_1, \dots, Z_n) = f_{2^\ell-1}(X_1, \dots, X_m)$	1 ... 1

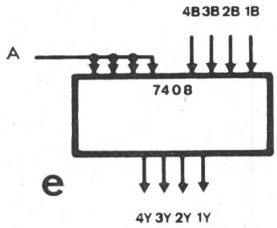
b



c

opération	description
AND	$(4Y, 3Y, 2Y, 1Y) = (4A \wedge 4B, 3A \wedge 3B, 2A \wedge 2B, 1A \wedge 1B)$

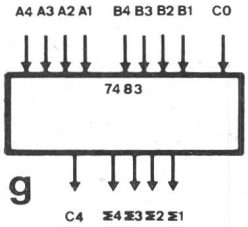
d



e

opération	description	A
ZERO	$(4Y, 3Y, 2Y, 1Y) = (0, 0, 0, 0)$	0
TRUE	$(4Y, 3Y, 2Y, 1Y) = (4B, 3B, 2B, 1B)$	1

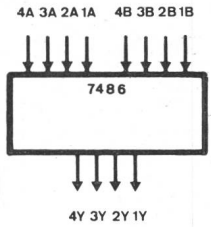
f



g

opération	description
ADD	$(C4, \Sigma 4, \Sigma 3, \Sigma 2, \Sigma 1) = (A4, A3, A2, A1) + (B4, B3, B2, B1) + (0,0,0,C0)$

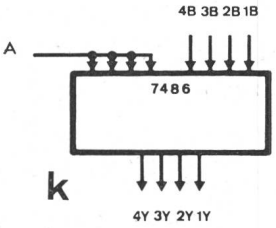
h



i

opération	description
XOR	$(4Y, 3Y, 2Y, 1Y) = (4A \oplus 4B, 3A \oplus 3B, 2A \oplus 2B, 1A \oplus 1B)$

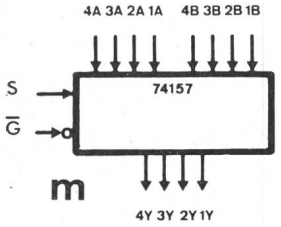
j



k

opération	description	A
TRUE	$(4Y, 3Y, 2Y, 1Y) = (4B, 3B, 2B, 1B)$	0
CPL	$(4Y, 3Y, 2Y, 1Y) = (\overline{4B}, \overline{3B}, \overline{2B}, \overline{1B})$	1

l

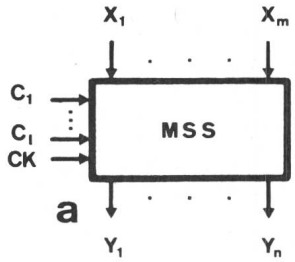


m

opération	description	$\overline{G}$	S
SELECT A	$(4Y, 3Y, 2Y, 1Y) = (4A, 3A, 2A, 1A)$	0	0
SELECT B	$(4Y, 3Y, 2Y, 1Y) = (4B, 3B, 2B, 1B)$	0	1
STROBE	$(4Y, 3Y, 2Y, 1Y) = (0, 0, 0, 0)$	1	Ø

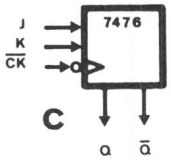
n

Fig. 3
Schémas et tables des opérations de composants séquentiels



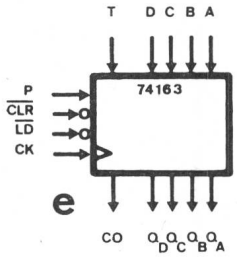
opération	description	$C_1 \dots C_\ell$
OP_0	$(Y_1, \dots, Y_n) \leftarrow f_0 \quad (X_1, \dots, X_m, Y_1, \dots, Y_n)$	0 ... 0
⋮		
$OP_{2^\ell-1}$	$(Y_1, \dots, Y_n) \leftarrow f_{2^\ell-1}(X_1, \dots, X_m, Y_1, \dots, Y_n)$	1 ... 1

b



opération	description	J	K
NOP	$Q \leftarrow Q$	0	0
CLEAR	$Q \leftarrow 0$	0	1
PRESET	$Q \leftarrow 1$	1	0
CPL	$Q \leftarrow \overline{Q}$	1	1

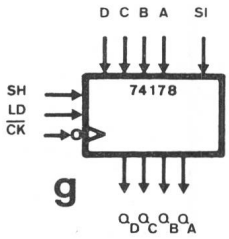
d



opération	description	$\overline{CLR}$	$\overline{LD}$	P
NOP	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	1	1	0
INC	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A) + (0, 0, 0, T)$	1	1	1
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D, C, B, A)$	1	0	Ø
CLEAR	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (0, 0, 0, 0)$	0	Ø	Ø

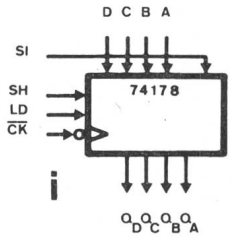
$CO = Q_D \wedge Q_C \wedge Q_B \wedge Q_A \wedge T$

f



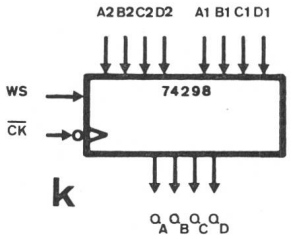
opération	description	SH	LD
NOP	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	0	0
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D, C, B, A)$	0	1
SHIFT	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, SI)$	1	Ø

h



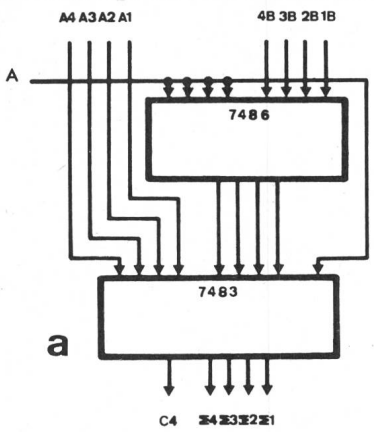
opération	description	SI	SH	LD
NOP	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	Ø	0	0
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (D, C, B, A)$	Ø	0	1
SHIFT 0	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, 0)$	0	1	Ø
SHIFT 1	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, 1)$	1	1	Ø

j



opération	description	WS
LOAD 1	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (A1, B1, C1, D1)$	0
LOAD 2	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (A2, B2, C2, D2)$	1

l



opération	description	A
ADD	$(C4, \Sigma4, \Sigma3, \Sigma2, \Sigma1) = (A4, A3, A2, A1) + (4B, 3B, 2B, 1B) + (0,0,0,0)$	0
SUB	$(C4, \Sigma4, \Sigma3, \Sigma2, \Sigma1) = (A4, A3, A2, A1) + (\overline{4B}, \overline{3B}, \overline{2B}, \overline{1B}) + (0,0,0,1)$	1

b

Fig. 4 Schéma et table des opérations d'un assemblage combinatoire

ment à 2 de ce nombre et résulte de l'interconnexion d'un additionneur à 4 bits (7483) et d'un opérateur OU-exclusif (7486). La table des opérations de cet assemblage apparaît à la figure 4b. Elle découle des tables des deux composants combinatoires utilisés (fig. 2h et 2l) et des relations d'interconnexion:

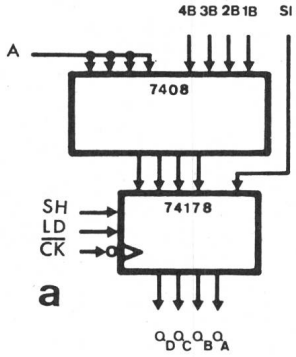
$$(B4, B3, B2, B1) = (4Y, 3Y, 2Y, 1Y) \quad (1)$$

$$CO = A \quad (2)$$

3.5 Assemblage séquentiel

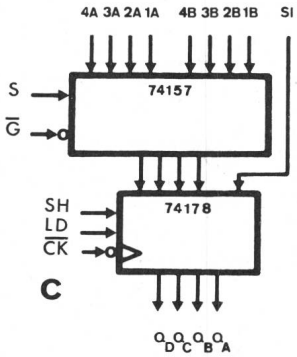
Un assemblage séquentiel à n bits est une machine séquentielle synchronisée à n sorties obtenue par l'interconnexion de composants combinatoires à n bits et d'un composant séquentiel à n bits. Cette machine est synchronisée par le signal d'horloge du composant séquentiel.

L'assemblage séquentiel (7408 + 74178) de la figure 5a résulte de l'interconnexion d'un opérateur ET (7408) et d'un



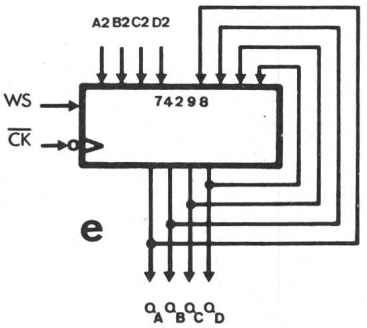
opération	description	A	SH	LD
NOP	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	Ø	0	0
CLEAR	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (0, 0, 0, 0)$	0	0	1
LOAD	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (4B, 3B, 2B, 1B)$	1	0	1
SHIFT	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, SI)$	Ø	1	Ø

b



opération	description	$\overline{G}$	S	SH	LD
NOP	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_D, Q_C, Q_B, Q_A)$	Ø	Ø	0	0
LOAD A	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (4A, 3A, 2A, 1A)$	0	0	0	1
LOAD B	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (4B, 3B, 2B, 1B)$	0	1	0	1
CLEAR	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (0, 0, 0, 0)$	1	Ø	0	1
SHIFT	$(Q_D, Q_C, Q_B, Q_A) \leftarrow (Q_C, Q_B, Q_A, SI)$	Ø	Ø	1	Ø

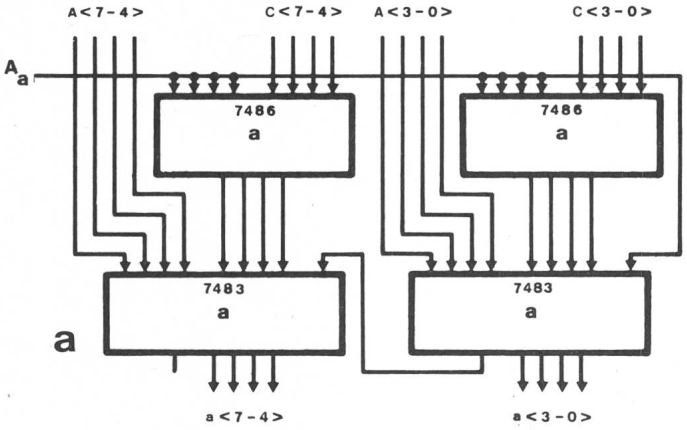
d



opération	description	WS
NOP	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (Q_A, Q_B, Q_C, Q_D)$	0
LOAD	$(Q_A, Q_B, Q_C, Q_D) \leftarrow (A2, B2, C2, D2)$	1

f

Fig. 5
Schémas et tables des opérations d'assemblages séquentiels



opération	description	A_a
OP_0	a = A + C	0
OP_1	a = A − C	1

b

Fig. 6 Schéma et table des opérations d'une ressource

registre à décalage (74178). La table des opérations de cet assemblage (fig. 5b) découle de la table du composant combinatoire (fig. 2f), de celle du composant séquentiel (fig. 3h) et de la relation d'interconnexion:

$$(D, C, B, A) = (4Y, 3Y, 2Y, 1Y) \quad (3)$$

La figure 5c présente un autre exemple d'assemblage séquentiel. La table des opérations de cet assemblage (fig. 5d) découle de la table du multiplexeur à une variable (74157) (fig. 2n), de celle du registre à décalage (74178) (fig. 3h) et de la relation d'interconnexion (3) précédente.

La figure 5e illustre un cas particulier d'assemblage séquentiel. Il s'agit d'un registre parallèle à 4 bits obtenu par rebouclage des sorties du multiplexeur à une variable avec mémorisation (74298) sur ses entrées A1, B1, C1 et D1. La table des opérations de ce registre parallèle à 4 bits (fig. 5f) se déduit de celle du multiplexeur à une variable avec mémorisation (74298) (fig. 3l) en vérifiant la relation:

$$(A1, B1, C1, D1) = (Q_A, Q_B, Q_C, Q_D) \quad (4)$$

3.6 Ressource

Une ressource de calcul à $k \times n$ bits ou, plus simplement, ressource à $k \times n$ bits est un système logique combinatoire à $k \times n$ sorties; elle est généralement réalisée à l'aide d'un ensemble de k composants ou assemblages combinatoires identiques à n bits.

La ressource a à 2×4 bits $2 \times (7483 + 7486)$ de la figure 6a se compose de deux additionneurs-soustracteurs à 4 bits (7483 + 7486). C'est un exemple de ressource dont la réalisation utilise deux assemblages combinatoires identiques. La table des opérations de cette ressource (fig. 6b) correspond à celle de l'assemblage combinatoire utilisé (fig. 4b). Le langage de description n'y est cependant plus employé au niveau des bits individuels mais à celui des mots, ce qui généralise cette table et la rend indépendante du nombre d'assemblages combinatoires de la ressource.

3.7 Registre

Un registre de transfert à $k \times n$ bits ou, plus simplement, registre à $k \times n$ bits est une machine séquentielle synchronisée à $k \times n$ sorties réalisée à l'aide d'un ensemble de k composants ou assemblages séquentiels identiques à n bits.

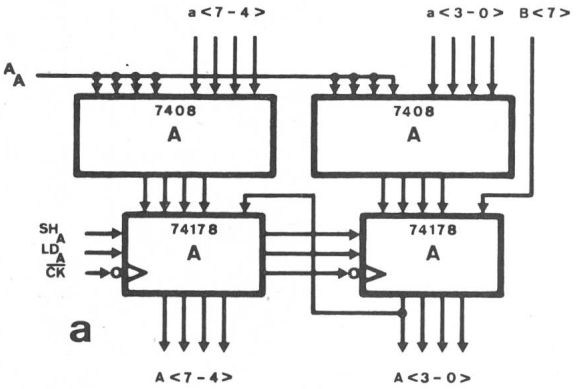
Le registre A à 2×4 bits $2 \times (7408 + 74178)$ de la figure 7a se compose de deux assemblages séquentiels (7408 + 74178). La table des opérations de ce registre (fig. 7b) correspond à celle de l'assemblage séquentiel utilisé (fig. 5b). Compte tenu du niveau de description employé, elle reste valable quel que soit le nombre d'assemblages séquentiels du registre.

4. Synthèse de l'unité de traitement

4.1 Méthode

La synthèse de l'unité de traitement d'un système digital réalisant un algorithme s'effectue en trois étapes:

– Déclaration des registres de l'unité de traitement et description fonctionnelle du système digital à l'aide d'un organigramme dans lequel le langage précédemment défini s'applique aux registres de l'unité de traitement.



opération	description	A_A	SH_A	LD_A
OP_0	NOP	Ø	0	0
OP_1	A ← 0	0	0	1
OP_2	A ← a	1	0	1
OP_3	SHL A ; A⟨0⟩ ← B⟨7⟩	Ø	1	Ø

b

Fig. 7 Schéma et table des opérations d'un registre

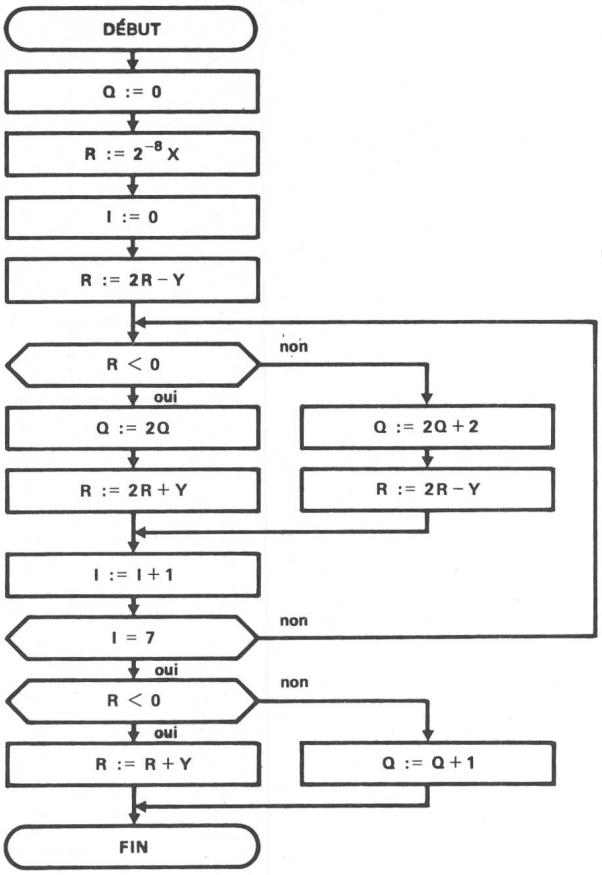


Fig. 8 Algorithme pour la division binaire de nombres entiers positifs de huit bits (X/Y = Q + R/Y)
Le symbole: = signifie «devient» (emprunté au langage PASCAL)

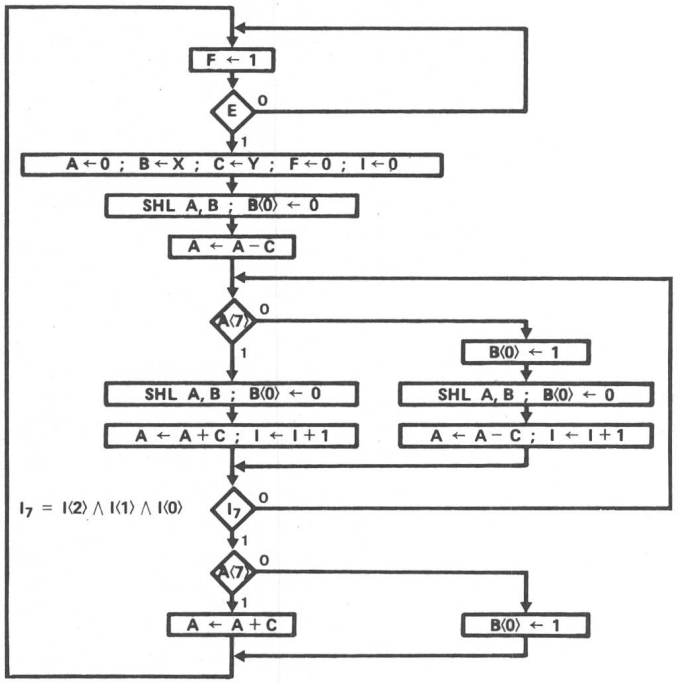


Fig. 9 Organigramme du système digital pour la division

– Construction du schéma de l'unité de traitement et déclaration de ses ressources.

– Réalisation de l'unité de traitement à l'aide des composants combinatoires et séquentiels disponibles sur le marché.

A titre d'exemple, les trois étapes de la méthode proposée sont appliquées à la synthèse de l'unité de traitement d'un système digital capable de diviser le nombre binaire X par le nombre binaire Y. Pour des nombres entiers positifs de 8 bits, le quotient Q et le reste R de cette division s'obtiennent conformément à l'algorithme de la figure 8 [4, pp. 190...199].

4.2 Déclaration des registres et description fonctionnelle du système digital

Le choix du nombre, de la dimension et de l'affectation des registres de l'unité de traitement ainsi que la description et le séquencement des transferts opérés entre ces registres à l'aide d'un organigramme sont effectués intuitivement sur la base de l'algorithme à exécuter.

Par exemple, un système digital capable de réaliser l'algorithme de la division (fig. 8) lorsque l'ordre d'exécution E est donné (E = 1), nécessite les registres suivants:

A < 1×8 > comme accumulateur pour les résultats intermédiaires et le reste R de la division,
B < 1×8 > pour le dividende X et le résultat final Q de la division,
C < 1×8 > pour le diviseur Y,
F < 1×1 > comme indicateur de fin de division,
I < 1×3 > comme registre d'indice,

et son fonctionnement est décrit par l'organigramme de la figure 9.

4.3 Construction du schéma et déclaration des ressources de l'unité de traitement

Le schéma de l'unité de traitement associe les ressources, relatives aux opérations arithmétiques et logiques qui interviennent dans l'algorithme, à l'ensemble des registres précédemment choisis et les interconnecte de façon à assurer tous les transferts d'information décrits dans l'organigramme.

Dans le schéma de l'unité de traitement destinée à la division (fig. 10), les ressources se réduisent à

a < 1×8 > pour l'addition et la soustraction du diviseur au résultat intermédiaire.

4.4 Réalisation de l'unité de traitement

La réalisation de l'unité de traitement consiste à choisir un ensemble de ressources et de registres capables d'effectuer les opérations décrites dans l'organigramme, puis à interconnecter leurs variables d'information conformément au schéma précédemment construit. L'unité de traitement ainsi réalisée

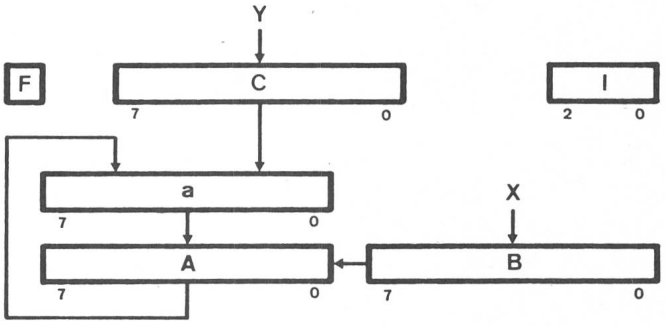
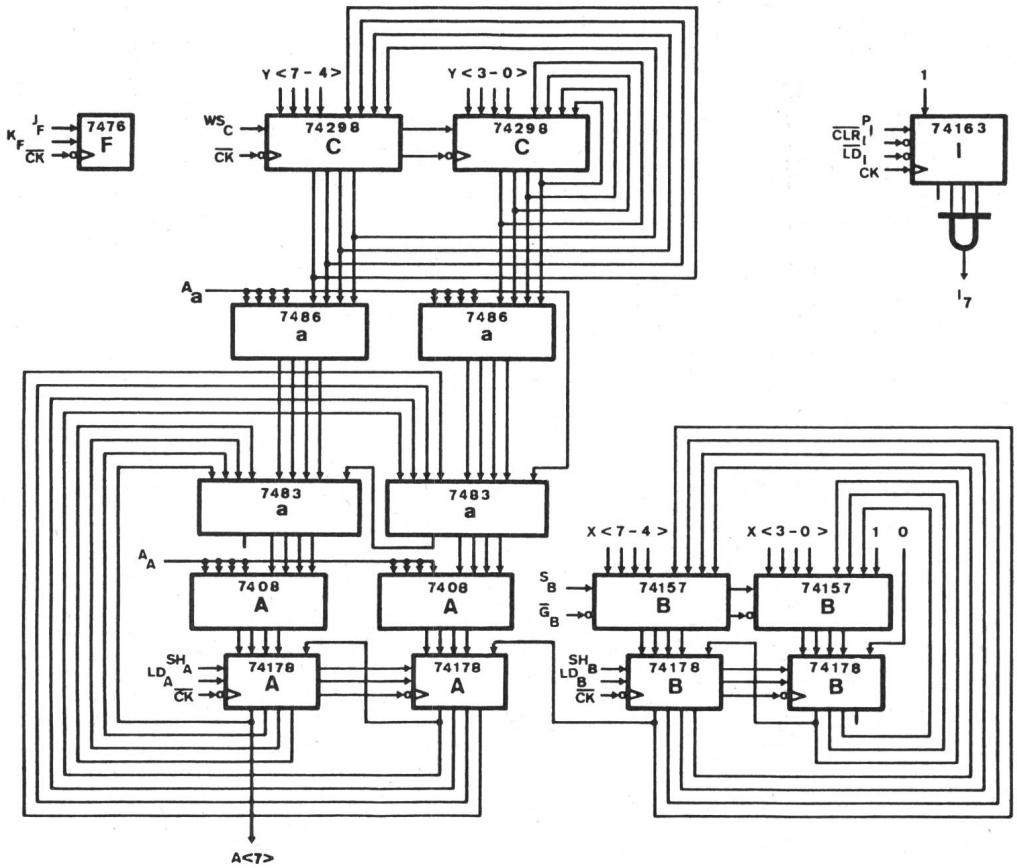


Fig. 10 Schéma de l'unité de traitement pour la division

Fig. 11
Réalisation de l'unité de traitement pour la division



est synchrone, car tous les registres reçoivent le même signal d'horloge CK.

La figure 11 propose une réalisation possible de l'unité de traitement destinée à la division. Cette réalisation résulte du choix d'une ressource et de registres strictement adaptés aux opérations de l'organigramme, par opposition à l'emploi de composants de nature plus universelle, comme des unités arithmétiques et logiques à 4 bits (74181) pour les opérations d'addition et de soustraction, par exemple.

5. Conclusion

La méthode de synthèse de l'unité de traitement d'un système digital repose sur l'emploi d'un langage de description. Elle utilise ce langage pour définir le fonctionnement du système digital dans un organigramme et conduit à une réalisation câblée de l'unité de traitement par interconnexion de composants combinatoires et séquentiels. Les composants combinatoires et séquentiels sont des circuits intégrés à moyenne échelle qui opèrent sur des ensembles de bits: les mots.

Bibliographie

[1] *M.M. Mano:* Computer system architecture. Englewood Cliffs/New Jersey, Prentice-Hall, 1976.

[2] *C. Piguet* e.a.: Le séquenceur d'un microprocesseur. Bull. ASE/UCS 70(1979)3, p. 126...132.

[3] *D. Mange:* Analyse et synthèse des systèmes logiques. St. Saphorin, Editions Georgi, 1978.

[4] *J.P. Hayes:* Computer architecture and organization. New York, McGraw-Hill, 1978.

[5] The TTL data book. Freising, Texas Instruments Deutschland GmbH, 1973.

Adresse de l'auteur

A. Stauffer, Chaire de Systèmes Logiques, EPFL, 16, chemin de Bellerive, 1007 Lausanne.