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: 70 (1979)

Heft: 3

Artikel: Le séquenceur d'un microprocesseur

Autor: Piguet, C. / Stauffer, A. / Perotto, J. F.

DOI: https://doi.org/10.5169/seals-905348

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: 15.07.2025

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

Le séquenceur d'un microprocesseur

Par C. Piguet, A. Stauffer, J.F. Perotto et J.J. Monbaron

681.325–181.48 : 681.326

Le séquenceur d'un microprocesseur est une machine séquentielle qui commande la séquence d'opérations élémentaires nécessaires à l'exécution d'une instruction du microprocesseur. Ces opérations élémentaires sont en grande partie des transferts d'informations entre registres, qu'il est possible de formaliser dans un langage dit «de transfert». La synthèse du séquenceur d'un microprocesseur du commerce est effectuée à partir de ce langage. Une décomposition en deux machines permet de réduire la complexité du séquenceur, permettant ainsi l'intégration du microprocesseur sur une seule puce.

Der Sequenzer eines Mikroprozessors ist eine sequentielle Maschine, die die Reihenfolge einzelner Schritte, die zur Ausführung eines Befehls dienen, steuert. Diese einzelnen Schritte bestehen zum grossen Teil aus Datenübertragungen von Registern zu Registern, die man formal in einer sogenannten «Registertransfer-Sprache» ausdrücken kann. Der Sequenzer eines kommerziellen Mikroprozessors ist auf der Basis einer solchen Sprache aufgebaut. Eine Zerlegung in zwei verschiedene Schaltungen reduziert die Komplexität des Sequenzers und ermöglicht dadurch die Integration des Mikroprozessors auf einem einzigen Baustein.

1. Introduction

L'utilisateur d'un microprocesseur ne connaît en général pas le fonctionnement interne de sa machine; cet article vise à combler cette lacune.

On sait qu'un microprocesseur se compose principalement de quatre types d'organes:

– les organes de traitement de l'information (unité arithmétique et logique),

– les organes de stockage de l'information (registres, flags),

– les organes de transport de l'information (bus d'adresses, bus de données),

– les organes de commande (séquenceur).

Si le fonctionnement de l'unité arithmétique et logique est relativement aisé à comprendre, il en va tout autrement pour le séquencement des instructions. En effet, l'exécution d'une instruction complexe requiert plusieurs opérations élémentaires dont la plupart sont des transferts d'information entre différents registres. Il est alors judicieux de décrire les instructions par un *langage de transfert*. Celui-ci est utilisé dans un organigramme pour représenter le fonctionnement interne du microprocesseur. Enfin, une synthèse effectuée à partir de cet organigramme permet une matérialisation du séquenceur ou unité de commande du microprocesseur.

2. Langage de transfert

Il existe plusieurs langages de transfert [1; 2]. On rappelle le langage simplifié qui sera utilisé.

2.1 Déclarations: Un registre ou une mémoire est repérée par des lettres majuscules, p.ex.:

RA:	registre d'adresse
IR < 1×8 >:	registre d'instructions de 8 bits
M < 32×8 >:	mémoire de 32×8 bits
A < 7 >:	le bit 7 de l'accumulateur
A < 4 – 0 >:	les bits zéro à quatre de l'accumulateur.

La numérotation des bits est effectuée de droite à gauche (fig. 1a).

2.2 Transferts: Le transfert est indiqué par une flèche, p.ex.:

A ← TB: le contenu du registre temporaire TB est chargé dans l'accumulateur A.

Le contenu de la position de la mémoire M adressée par RA est noté M(RA). Ainsi:

M(RA) ← A: le contenu de A est chargé dans la position mémoire adressée par RA.

2.3 Opérateurs: Les opérateurs sont

«+»:	somme arithmétique
«–»:	différence arithmétique
«∧»:	et logique
«∨»:	ou logique
«$\overline{A}$»:	inversion logique
«(+)»:	ou exclusif

2.4 Décalages: Les opérations de décalage sont

ROTL:	rotation à gauche de 1 bit (fig. 1b)
ROTR:	rotation à droite de 1 bit (fig. 1c)
ROTL A, F:	rotation de registres accolés (fig. 1d)

2.5 Incrémentation-décrémentation: Ces opérations sont

PC ← PC + 1:	incrémentation du compteur de programme PC
SP ← SP – 1:	décrémentation du pointeur de pile SP.

2.6 Simultanéité

Deux opérations simultanées sont représentées sur la même ligne, par exemple:

A ← C; B ← D: transfert du contenu de C et D dans A et B respectivement.

3. Architecture d'un microprocesseur

L'architecture du microprocesseur dont on veut synthétiser le séquenceur est représentée à la figure 2. C'est en fait celle du microprocesseur 8008 de INTEL [3]. Cette dernière diffère néanmoins en ce que les registres PCH et PCL du compteur de programme font partie des mémoires STH et STL de la pile, de même que l'accumulateur A constitue l'un des registres de la mémoire de travail R (scratch-pad).

Le BUS interne de 8 bits est relié au registre de sortie BU et à deux registres temporaires TA et TB qui sont les registres d'entrée de l'unité arithmétique et logique ALU. La sortie de l'ALU est connectée au BUS. Les six registres B, C, D, E, H et L de la mémoire R, adressés par le pointeur RP, permettent

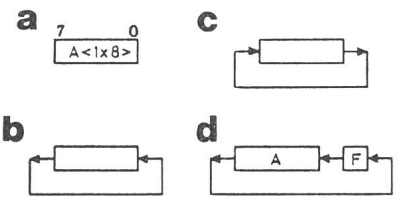

Fig. 1
Registres et décalages

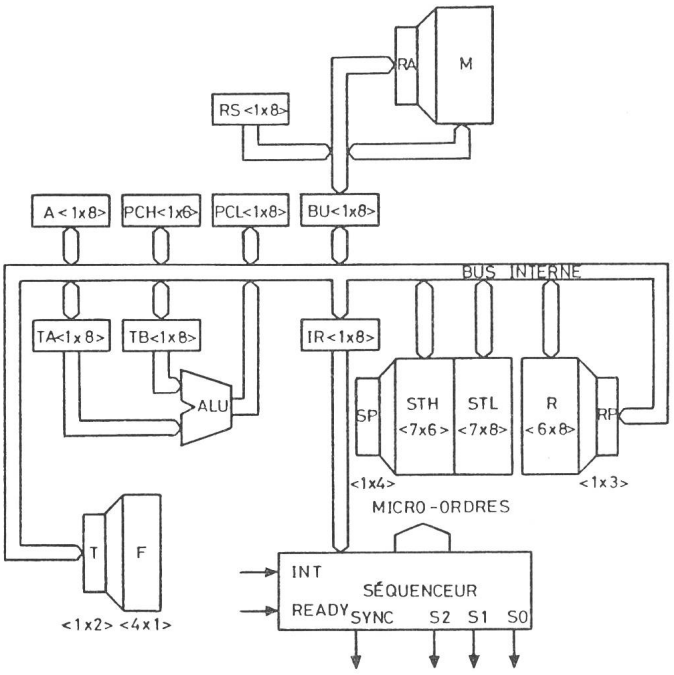

Fig. 2 Architecture du microprocesseur

code octal	mnémonique	description
3 d s	LOAD d, s	R(d) ← R(s)
3 d 7	LOAD d,(HL)	R(d) ← M(HL)
3 7 s	LOAD(HL), s	M(HL) ← R(s)
0 d 6 / n	LOAD d,# n	R(d) ← M(PC+1)
0 7 6 / n	LOAD(HL),# n	M(HL) ← M(PC+1)

a

d, s	registre
0	A
1	B
2	C
3	D
4	E
5	H
6	L

b

Fig. 3 Instructions de chargement

de stocker des données ou des adresses. Quatre indicateurs d'états F (flags) sont adressés par le pointeur T, et sont testés lors d'instructions conditionnelles. Ces indicateurs sont ceux du report F(0) (carry), du résultat nul F(1), du signe F(2) et de parité F(3). Le PC est connecté au BUS, de même que la pile ST composée de 7 registres de 14 bits permettant ainsi sept niveaux de sous-routines. Le registre d'instruction IR commande le séquenceur qui délivre les micro-ordres nécessaires au séquencement des instructions. La mémoire M (ROM, RAM) et le registre RA sont extérieurs au microprocesseur. Enfin, un registre externe RS peut se substituer à la mémoire M en cas d'interruption, permettant de charger une instruction dans le registre IR (en général un restart).

4. Instructions du microprocesseur

Pour effectuer la tâche qui lui est assignée, un microprocesseur doit être programmé. A cet effet, le constructeur fournit à l'utilisateur la liste d'instructions que le microprocesseur est capable d'exécuter. Cette liste est indispensable pour effectuer la synthèse du séquenceur du microprocesseur.

Sous sa forme élémentaire, une instruction se présente comme un mot binaire formé de 1 et de 0, par exemple 00011010. Afin de faciliter sa lecture, le mot binaire peut être décomposé en groupes de trois bits en partant de la droite, p.ex. 00 011 010. Chacun de ces groupes correspond à un chiffre dans le système de numérotation de base huit (code octal), l'instruction s'écrit alors 032. Pour faciliter encore la tâche du programmeur, une instruction peut s'écrire avec des symboles qui rappellent sa nature (mnémonique); par exemple l'instruction 032 est représentée par RRC A (fig. 4).

Les figures 3 à 7 présentent la liste d'instructions de l'INTEL 8008 [3], chacune d'entre elles étant décrite par son code octal, son mnémonique tiré de [4], et sa description utilisant le langage de transfert.

Le code octal peut comporter des zones symboliques, telles que

- n représentant une donnée,
- m représentant une adresse,
- rrr représentant une adresse d'entrée/sortie,
- Z représentant une opération arithmétique ou logique (Fig. 5b),
- t représentant une condition de test (Fig. 6b),
- d, s représentant une adresse de registre (Fig. 3b).

C'est au moment de l'écriture du programme que les zones symboliques sont remplacées par leurs valeurs numériques.

4.1 Instructions de chargement (fig. 3)

Les descriptions montrent que la mémoire R est adressée par les valeurs d ou s et la mémoire M par le contenu des registres HL ou PC. Par exemple M (PC+1) repère la position de la mémoire M adressée par le PC+1, c'est-à-dire le byte suivant l'instruction en cours dans le programme. Le symbole #n dans LOAD d, #n indique qu'il s'agit du chargement d'une valeur immédiate, c'est-à-dire mémorisée dans le programme juste après l'instruction.

4.2 Instructions d'incrémentation-décrémentation et de décalages (fig. 4)

Les indicateurs F(1), F(2), et F(3) sont mis à jour par les deux premières instructions, tandis que les quatre dernières affectent le report F(0) (carry).

code octal	mnémonique	description
0 d 0	INC d	R(d) ← R(d)+1
0 d 1	DEC d	R(d) ← R(d)−1
0 0 2	RL A	ROTL A; F(0) ← A <7>
0 1 2	RR A	ROTR A; F(0) ← A <0>
0 2 2	RLC A	ROTL A, F(0)
0 3 2	RRC A	ROTR A, F(0)

Fig. 4 Instructions d'incrémentation/décrémentation et de décalages

4.3 Instructions arithmétiques et logiques

Elles sont rassemblées selon la figure 5a. où les symboles Z, ALU, *, Y sont expliqués dans la figure 5b, selon la valeur de Z. Tous les indicateurs sont mis à jour par ces instructions. Lors des comparaisons (Z = 7), le transfert n'est pas effectué.

4.4 Instructions de branchement (fig. 6)

Lors de l'instruction RST a, c'est l'adresse octale 00 0 a 0 qui est forcée dans le PC, ce qui permet d'attribuer des blocs de 8 bytes à chaque restart.

4.5 Instructions machine et d'entrée/sortie (fig. 7)

Pour les instructions d'entrée/sortie de la figure 7b, le symbole $ r dans LOAD A, $ r indique qu'il s'agit du chargement d'une valeur dans un périphérique repéré par le code r.

5. Séquencement

Un cycle-mémoire correspond à une lecture/écriture en mémoire. Celui-ci comporte deux phases, à savoir une phase de recherche de l'information en mémoire (fetch) et une phase d'exécution (execute). Une instruction peut comporter plusieurs cycles-mémoires. Pour chaque instruction, la phase de recherche du premier cycle-mémoire est toujours un transfert du code opératoire dans le registre d'instruction. Toutes les instructions faisant référence à la mémoire pour aller chercher une donnée comportant un ou deux bytes nécessitent des cycles-mémoires supplémentaires. Il en va de même pour les instructions qui comportent plusieurs bytes, puisqu'il faut au moins autant de cycles-mémoires qu'il y a de bytes. L'exécution d'une instruction requiert donc une succession de plusieurs opérations élémentaires. En utilisant le langage de transfert et compte tenu de la structure du microprocesseur, nous représentons le séquencement des opérations élémentaires d'une ou de plusieurs instructions par un organigramme.

5.1 Exemple d'instructions à un cycle

L'instruction HLT provoque le passage du séquenceur à l'état STOP. Seule l'apparition du signal extérieur INT permet d'en sortir. L'instruction LOAD d,s effectue un transfert d'information entre deux registres internes, à savoir R(d) ← R(s). L'organigramme correspondant à ces instructions est représenté à la figure 8. On constate qu'il faut une séquence de 9 opérations élémentaires pour exécuter l'instruction LOAD d,s. Les quatre premières opérations élémentaires transfèrent dans le registre de sortie BU les deux bytes du PC et effectuent son incrémentation. Le contenu du registre BU est transféré en deux fois dans le registre d'adresse RA de la mémoire M. L'instruction lue dans la mémoire M adressée par RA est alors transférée dans le registre d'entrée BU. Ces deux transferts sont gérés de manière externe. Le signal extérieur READY, apparaissant dès que l'information est présente dans le registre BU, permet de synchroniser la mémoire et le microprocesseur. Le contenu du registre BU est alors transféré dans le registre d'instruction IR. Les quatre dernières opérations élémentaires permettent de charger un registre par un autre en utilisant le registre TB comme registre temporaire.

code octal	mnémonique	description
1 X 4 / m	JUMP m	PCL ← M(PC+1) ; PCH ← M(PC+2) <5 – 0>
1 t 0 / m	JUMP ,t m	PCL ← M(PC+1) ; PCH ← M(PC+2) <5 – 0> conditionnel à F(t)
1 X 6 / m	CALL m	STL(SP) ← PCL ; STH(SP) ← PCH PCL ← M(PC+1) ; PCH ← M(PC+2) <5 – 0> ; SP ← SP+1
1 t 2 / m	CALL ,tm	STL(SP) ← PCL ; STH(SP) ← PCH PCL ← M(PC+1) ; PCH ← M(PC+2) <5 – 0> ; SP ← SP+1 conditionnel à F(t)
0 X 7	RET	PCL ← STL(SP) ; PCH ← STH(SP) SP ← SP – 1
0 t 3	RET , t	PCL ← STL(SP) ; PCH ← STH(SP) SP ← SP – 1 conditionnel à F(t)
0 a 5	RST a	STL(SP) ← PCL ; STH(SP) ← PCH PCL ← 0a0 ; PCH ← 00

a X: don't care

t	condition
0	F(0) = 0
1	F(1) = 0
2	F(2) = 0
3	F(3) = 0
4	F(0) = 1
5	F(1) = 1
6	F(2) = 1
7	F(3) = 1

b

Fig. 6 Instructions de branchement

code octal	mnémonique	description
2 Z S	ALU A , s	A ← A * R(s) ; Y
2 Z 7	ALU A ,(HL)	A ← A * M(HL) ; Y
0 Z 4 / n	ALU A ,# n	A ← A * M(PC+1); Y

a

Z	ALU	*	Y
0	ADD	+	
1	ADDC	+ F(0) +	
2	SUB	–	
3	SUBC	– F(0) –	
4	AND	∧	F(0) ← 0
5	XOR	⊕	F(0) ← 0
6	OR	∨	F(0) ← 0
7	COMP	–	

b

Fig. 5 Instructions arithmétiques et logiques

code octal	mnémonique	description
0 0 X / 3 7 7	HLT	aller dans l'état STOP jusqu'à l'apparition du signal d'interruption INT

a

code binaire	mnémonique	description
01 00r rr1	LOAD A , $ r	A ← BU
01 rrr rr1	LOAD $r, A	BU ← A

b

Fig. 7 Instructions machine et d'entrée/sortie

5.2 Exemple d'une instruction à deux cycles

L'organigramme de la figure 9 représente le séquencement de l'instruction à deux cycles-mémoires ADD A, # n. On peut constater que les cinq premières opérations élémentaires sont identiques à celles de l'organigramme de la fig. 8. Elles correspondent à la recherche de l'instruction. L'instruction est ensuite décodée. Etant une instruction à deux bytes, une lecture du deuxième byte est exécutée de la même manière que la lecture du premier byte. Ce deuxième byte est chargé dans le registre TB et additionné à l'accumulateur A. Les deux lectures en mémoire sont effectuées par les mêmes opérations élémentaires.

A la fin de l'exécution d'une instruction, le signal INT d'interruption est testé. S'il est inactif, la recherche de la prochaine instruction est effectuée. Dans le cas contraire, une procédure d'interruption est assurée.

5.3 Matérialisation

L'organigramme de la figure 9 pourrait être complété de façon à tenir compte de toutes les instructions du microprocesseur. Il peut alors être matérialisé par une machine séquentielle synchronisée [5] ou une machine microprogrammée [2, pp. 335...340]. Pour réduire la compléxité du séquenceur, il est avantageux de le décomposer en deux machines séquentielles synchronisées.

Une première machine effectue le séquencement des opérations élémentaires d'un cycle-mémoire (ou cycle-machine) qui consiste en une phase de lecture/écriture en mémoire et une phase d'exécution. Une deuxième machine effectue le séquencement des différents cycles-mémoires que peut comporter une instruction du microprocesseur.

6. Le séquenceur

Le microprocesseur INTEL 8008 est muni d'un répertoire d'instructions ayant jusqu'à trois cycles-mémoires M_1, M_2 et M_3, chacun de ceux-ci pouvant comporter de 3 à 5 états T_1, T_2, T_3, T_4 et T_5.

6.1 Générateur d'états

La machine séquentielle synchronisée effectuant le séquencement des états d'un cycle-mémoire comporte 8 états. Les états T_1 et T_2 permettent la lecture/écriture en mémoire, les états T_3 à T_5 de décoder puis d'exécuter des instructions. En outre, il existe un état WAIT, un état STOP et un état T_1 I atteint lors d'une reconnaissance d'interruption. La figure 10 représente le graphe de cette machine séquentielle synchronisée. Les transitions entre les états de ce graphe sont conditionnées par des équations booléennes où les variables apparaissent sous forme vraie et complémentaire. Les variables d'entrée sont:

- les signaux externes READY et INT
- les signaux Z_0 à Z_6 indiquant le type d'instruction, tel que HLT, LOAD d,s, ADD A, # n, etc. ...
- les états M_1, M_2 et M_3 indiquant dans quel cycle-mémoire se trouve le microprocesseur.

Pour cette machine, on distingue un comportement asynchrone [7, p. 197] où l'état de sortie ne dépend pas de la durée des états d'entrée, mais seulement de leur ordre de succession, d'un comportement synchrone [7, p. 220] où deux états d'entrée identiques successifs sont distincts. Dans le premier cas, la notation du grafcet [6] est conforme à la figure 11a alors que dans le deuxième cas, la notation d'un graphe classique est adoptée selon la figure 11b.

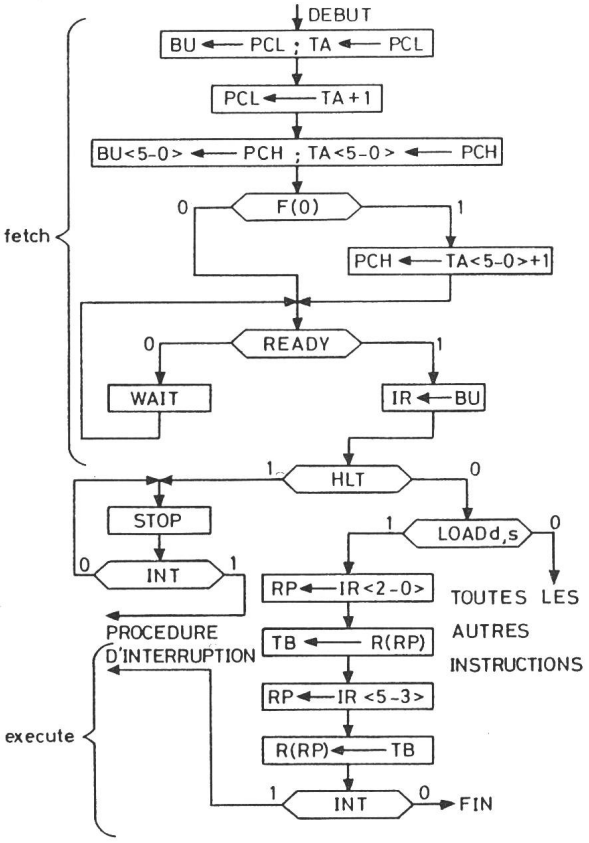

Fig. 8 Organigramme de HLT et LOAD d, s

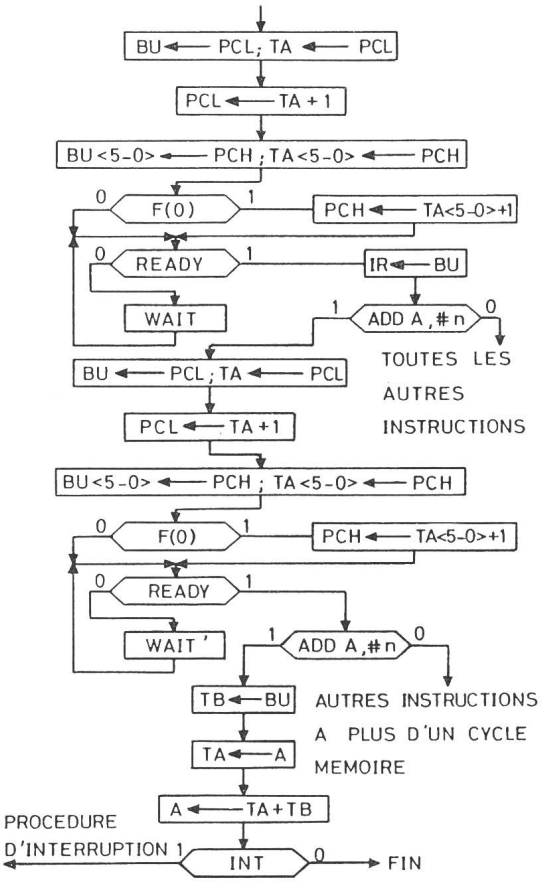

Fig. 9 Organigramme de ADD A, # n

Les variables Z_0 à Z_6 de la figure 10 sont définies de la manière suivante:

Z_0 = HLT

Z_1 = RET t(CF)[1]), soit une instruction à 1 cycle M_1 et à 3 états T_1, T_2 et T_3.

Z_2 = LOAD d,(HL) + LOAD d, # n + LOAD (HL), # n + ALU A,(HL) + ALU A, # n + JUMP m + JUMP,t m + CALL m + CALL,t m + LOAD A, $ r + LOAD $ r,A, soit les instructions à plus d'un cycle dont le premier M_1 a 3 états T_1, T_2 et T_3.

Z_3 = LOAD $ r,A + LOAD (HL), s, soit les instructions à 2 cycles M_1 et M_2 et 3 états T_1, T_2 et T_3 pour le second cycle.

Z_4 = LOAD (HL), # n + JUMP m + JUMP,t m + CALL m + CALL,t m, soit les instructions à 3 cycles M_1, M_2 et M_3 dont le second a 3 états T_1, T_2 et T_3.

Z_5 = LOAD (HL), # n + JUMP,t m(CF)[1]) + CALL,t m(CF)[1]), soit les instructions à 3 cycles M_1, M_2 et M_3 dont le troisième a 3 états T_1, T_2 et T_3.

Z_6 = LOAD (HL),s, soit une instruction à plus d'un cycle dont le premier M_1 a 4 états T_1, T_2, T_3 et T_4.

Le passage de l'état T_1 ou T_1 I à T_2 a toujours lieu. Si le signal READY est présent, l'état T_3 est atteint, sinon la machine reste dans l'état WAIT jusqu'à l'apparition de ce signal. Si l'instruction HLT est décodée dans le cycle M_1 à l'état T_3, la machine entre dans l'état STOP d'où seul le signal INT lui permet de sortir. On constate que les instructions conditionnelles RET, JUMP et CALL sont terminées à l'état T_3 lorsque le branchement n'est pas effectué.

L'état de sortie peut être codé par les trois variables S_0, S_1 et S_2 comme indiqué dans [3], ce qui définit les 8 états du graphe de la figure 10.

[1]) Condition non réalisée.

6.2 Générateur de cycles

La machine séquentielle synchronisée effectuant le séquencement des cycles-mémoires d'une instruction comporte 3 états M_1, M_2 et M_3. Les variables d'entrée de cette machine sont les états T_1 à T_5 et les signaux Z_2 à Z_6 indiquant le type d'instruction.

La figure 12 représente le grafcet [6] de cette machine. A l'état M_1, la machine teste si l'instruction nécessite au moins deux cycles, et atteint l'état M_2 en cas de test positif. A l'état M_2, la machine teste si l'instruction est terminée, auquel cas elle revient à M_1, ou si elle nécessite trois cycles, auquel cas elle atteint l'état M_3. Elle revient à l'état M_1 depuis l'état T_3 ou T_5 selon le type d'instruction.

L'état de sortie peut être codé à l'aide de deux variables Q_1 et Q_2.

6.3 Structure du séquenceur

La structure du séquenceur est représentée à la figure 13. Elle correspond au schéma détaillé du constructeur [8, p. 77]. Le code opératoire de l'instruction stockée dans le registre d'instruction IR est décodé par le circuit logique combinatoire appelé «décodeur d'instructions». Les deux machines séquentielles synchronisées, soit le «génerateur d'états» et le «générateur de cycles», fournissent les codes $S_0S_1S_2$ et Q_1Q_2 correspondant respectivement aux états et aux cycles. Elles sont synchronisées par le signal d'horloge SYNC délivré par le circuit d'horloge. La figure 14 représente le chronogramme des signaux SYNC, H_1 et H_2. Un circuit logique combinatoire, dit «générateur de micro-ordres», délivre les signaux qui commandent les opérations élémentaires. Pour un état codé par $S_0S_1S_2$, un cycle codé par Q_1Q_2, et une instruction donnée, deux micro-ordres synchronisés, le premier par H_1 et le second

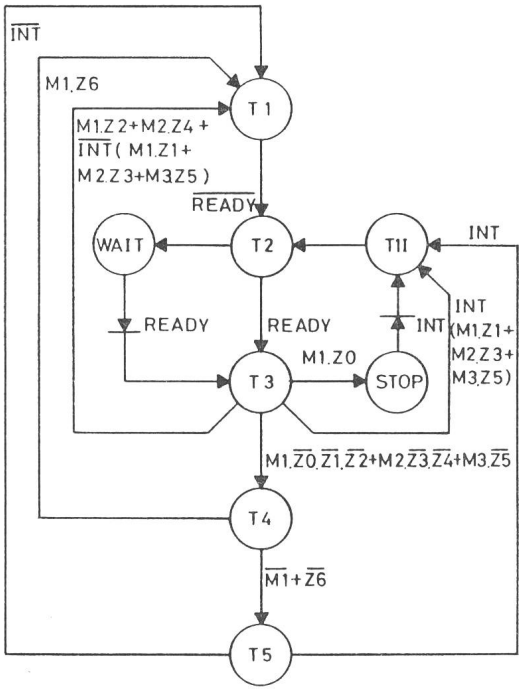

Fig. 10 Graphe du générateur d'états

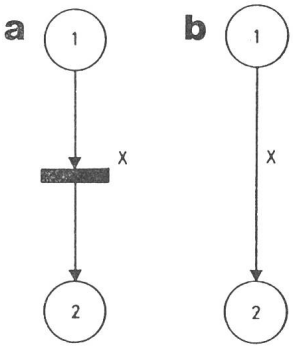

Fig. 11 Comportement asynchrone et comportement synchrone

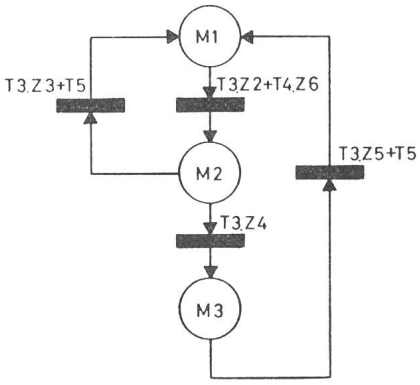

Fig. 12 Grafcet du générateur de cycles

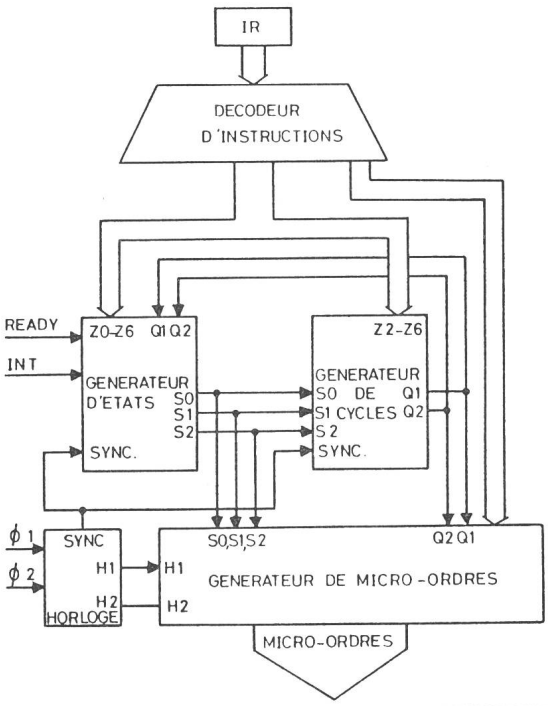

Fig. 13 Structure du séquenceur

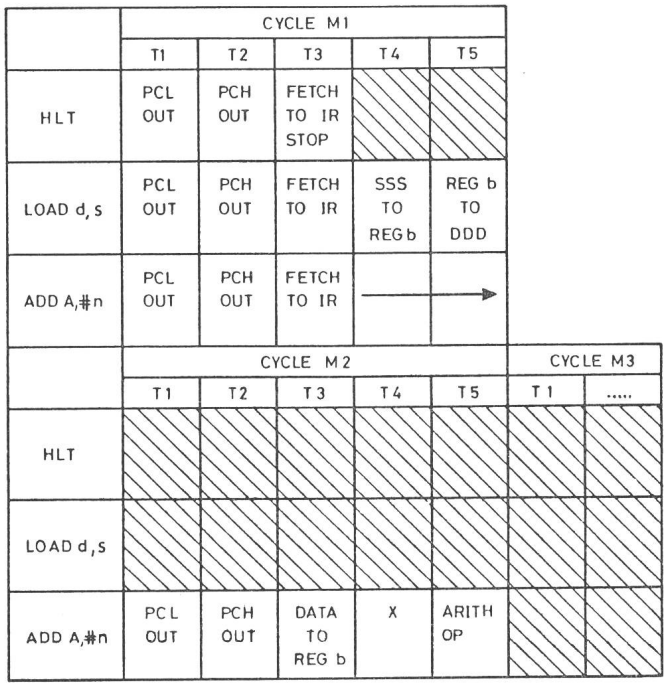

Fig. 15 Représentation condensée

par H_2, peuvent être donnés. Ainsi, pour l'instruction LOAD d, s du § 5.1, les deux premiers micro-ordres de la figure 8 sont donnés à l'état T_1, les deux suivants à l'état T_2, le cinquième à l'état T_3 et les quatre derniers aux états T_4 et T_5.

7. Notation du constructeur

Comme il est difficile de représenter le séquencement des opérations élémentaires nécessaires à l'exécution de toutes les instructions au moyen d'organigrammes, le constructeur propose une représentation très condensée [3], selon la figure 15, qui correspond aux instructions HLT, LOAD d, s et ADD A, # n.

Les indications «PCL out» et «PCH out» doivent être comprises comme le transfert du PC en deux fois au registre BU. L'indication «FETCH TO IR» doit être comprise comme le chargement de IR par l'instruction. Les indications «SSS TO REG b» et «REG b TO DDD» déterminent les transferts d'un registre adressé par s dans TB, puis TB dans un registre adressé par d. L'indication «DATA TO REG b» signifie le transfert du registre BU dans TB, l'indication «X» un état inactif et «ARITH OP» l'exécution de l'addition.

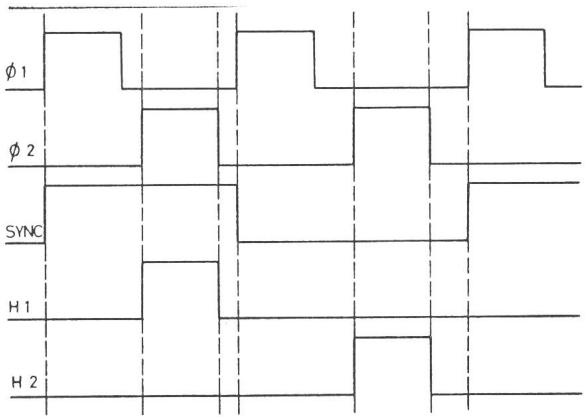

Fig. 14 Chronogramme du circuit d'horloge

8. Interruption

Il est encore intéressant d'analyser le comportement du séquenceur en cas d'interruption. La figure 16 propose l'organigramme correspondant. Lorsque le test du signal INT est positif, le séquenceur atteint l'état T_1 I où PCL n'est pas incrémenté de même que PCH puisque le report $F(0) = 0$ dans tous les cas. Le contenu du registre externe RS est alors transféré dans le registre BU. Ce transfert est géré de manière externe.

Puis il y a transfert du registre BU dans le registre IR. L'instruction forcée peut être quelconque, mais en général c'est

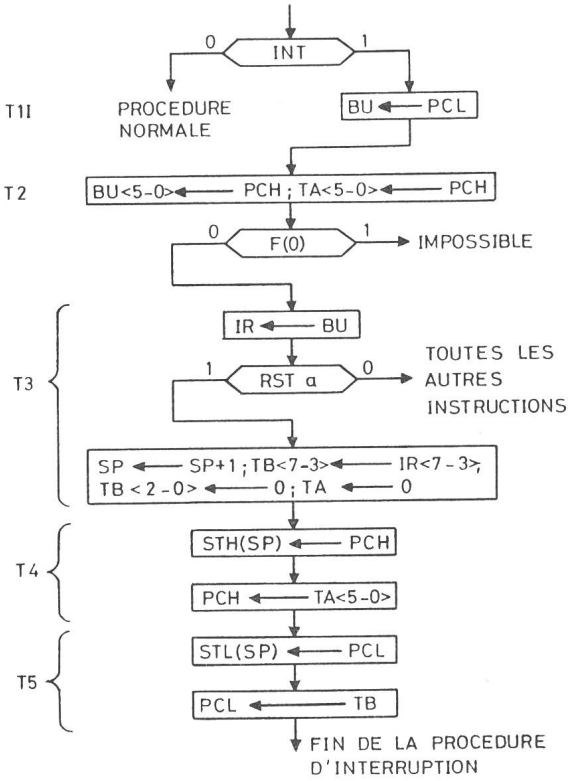

Fig. 16 Procédure d'interruption

une instruction de restart. Celle-ci est exécutée de manière à charger 0 0 dans TA et 0 a 0 dans TB. En outre, le pointeur de pile SP est incrémenté de manière à sauver en deux fois le PC dans les registres de pile ST. Ensuite, les registres TB et TA sont chargés dans le PC.

9. Conclusion

Le choix du microprocesseur INTEL 8008 pour illustrer la synthèse de séquenceurs a été fait d'une part parce qu'il est le plus simple microprocesseur à 8 bits et d'autre part parce que la qualité de la documentation fournie concernant le séquencement des instructions est excellente [3].

Pratiquement, nous ignorons quelle matérialisation obtenue à partir des graphes des figures 10 et 12 a été retenue par le constructeur. Le séquenceur du 8008 occupe environ un tiers de la surface de la puce du circuit intégré.

L'architecture du 8080 [9] est différente, puisque le BUS d'adresses est dinstinct du BUS de données. Néanmoins, son séquenceur pourrait être synthétisé selon la même méthode.

Cet article montre à l'utilisateur qu'une instruction machine, qui lui paraît extrêmement rudimentaire, est en fait une séquence de plusieurs opérations élémentaires, et de ce fait, beaucoup plus complexe qu'il ne l'imaginait.

Bibliographie

[1] *F. J. Hill* and *G. R. Peterson:* Digital systems: Hardware, organization and design. New York, John Wiley, 1973.

[2] *Y. Chu:* Computer organization and micro-programming. Englewood Cliffs/N. J., Prentice-Hall, 1972.

[3] Intel-MCS-8 user's manual. Santa Clara/California, Intel Corporation, 1974.

[4] *J. D. Nicoud:* Microprocessor 8080: Software. Microscope 1(1976)4, p. 11...18.

[5] *C. Clare:* Designing logic systems using state machines. New York, McGraw-Hill, 1973.

[6] Rapport de la commission de normalisation de la représentation du cahier des charges d'un automatisme logique (groupe de travail AFCET, systemes logiques). Automatisme 23(1978)2, p. 66...83.

[7] *D. Mange:* Analyse et synthèse des systèmes logiques. Traité d'électricité. Vol. V. Saint Saphorin, Editions Georgi, 1978.

[8] *D. R. McGlynn:* Microprocessors. Technology, architecture and applications. New York, John Wiley, 1976.

[9] *B. Soucek:* Microprocessors and microcomputers. New York, John Wiley, 1976, pp. 251...298.

Adresses des auteurs

C. Piguet et J. F. Perotto, Centre Electronique Horloger S. A., rue Breguet 2, 2000 Neuchâtel;

A. Stauffer, Chaire de Systèmes Logiques de l'EPFL, 16, Chemin de Bellerive, 1007 Lausanne;

J. J. Monbaron, Institut de Microtechnique de l'Université de Neuchâtel, rue de Pierre-à-Mazel 7, 2000 Neuchâtel.

Literatur – Bibliographie

DK: 621.317.3-501 : 681.32 SEV-Nr. A 715

Gütekriterien für Messeinrichtungen. Von *Pjotr Vassiljevic Novickij.* Berlin, VEB Verlag Technik, 1978; 8°, 151 S., 83 Fig., 6 Tab. Preis: gb. DDR-M 20.–.

Recht lange konnte man sich in der Messtechnik damit begnügen, Fehlerbetrachtungen für ein Messergebnis rudimentär anhand der Fehlerklasse (Fehler vom Endwert) des oder der verwendeten Messgeräte durchzuführen. Für den möglichen Fehler wurde allgemein eine «worst case»-Analyse mit algebraischer Addition der Einzelfehler durchgeführt. Ähnlich wurde häufig auch die Zuverlässigkeit von Meßsystemen aufgrund der Zuverlässigkeit der einzelnen Subsysteme bestimmt.

Bei modernen Meßsystemen mit ihrer Vielzahl an Komponenten und Verknüpfungen sind derart simplifizierende Untersuchungen nicht mehr zulässig. Fundierte Aussagen erfordern wesentlich tiefergehende statistische und auch informationstheoretische Untersuchungen, welche nicht nur den möglichen Fehler beschreiben, sondern auch die Wahrscheinlichkeit seines Auftretens. Es ist das Ziel des vorliegenden, von mehreren Autoren gemeinsam bearbeiteten Werkes, dem Leser die hierfür notwendigen Grundlagen in einer auch didaktisch ansprechenden Weise nahezubringen.

Zu diesem Zwecke werden nach einer Einführung in den Begriff des Fehlers zuerst in knapper und prägnanter Form sowohl die statistischen als auch die informationstheoretischen Grundlagen zur Beschreibung der Fehler und der Wahrscheinlichkeit ihres Auftretens erläutert. Bereits dieses Kapitel macht das Buch lesenswert, liefert es doch in konziser und verständlicher Form die Grundsätze zu einer wesentlich allgemeiner verwendbaren stichprobenmässigen Qualitätsuntersuchung. Auch die häufig schwierig verständlichen Begriffe der Informationstheorie werden sehr anschaulich erläutert.

Anschliessend werden, auf diesen Grundlagen aufbauend, eine Vielzahl häufig verwendeter fundamentaler Meßsysteme mit aller notwendigen Strenge und Tiefe behandelt.

Ein letztes Kapitel ist dem Versuche gewidmet, eine Beschreibungsgrösse für die *Qualität* eines Messgerätes zu definieren, wobei die Autoren sich vollauf bewusst sind, dass hier nur Denkanstösse, nicht aber definitive Lösungen vermittelt werden können. Der Rezensent kann dieses Werk, das auf engem Raum einen sehr abgerundeten Einblick in einen komplexeren Problemkreis gibt, nicht nur dem Spezialisten der Messgeräteherstellung empfehlen. Auch für einen grossen Anwenderkreis dürfte es hilfreiche Informationen zur Beurteilung der Güte und Brauchbarkeit von Meßsystemen liefern. *J. Weiler*

DK: 621.313 : 621.314 SEV-Nr. A 712

Baustein elektrische Maschine. Von *Thomas Keve* und *Helmut Roeloffzen.* Stuttgart, Verlag Berliner Union/Stuttgart u. a., Verlag W. Kohlhammer, 1978; 8°, 280 S., Fig., Tab. – Lehrbuchreihe Elektronik, Band 5 – kart. DM 39.80.

Das vorliegende Lehrbuch umfasst hauptsächlich das Gebiet der Anwendung elektrischer Maschinen. Entsprechend dem Titel werden die Maschinen als Bausteine vorgestellt und die zum Verständnis ihrer Wirkungsweise und ihrer Einsatzmöglichkeiten notwendigen Kenntnisse vermittelt.

Im ersten Teil werden die vielfältigen Einsatzbedingungen der elektrischen Maschinen sowie die durch die Betreiber an sie gestellten Anforderungen behandelt. Im zweiten Abschnitt werden in leicht verständlicher Form der prinzipielle Aufbau, die Wirkungsweise sowie das Betriebsverhalten der elektrischen Maschinen wie auch des Transformators beschrieben. Eine grosse Anzahl gut ausgewählter Bilder zeigt dabei, dass auf eine praxisnahe Darstellung Wert gelegt wurde. Im dritten Teil werden tiefer gehende theoretische Kenntnisse über die Wirkungsweise und das Betriebsverhalten vermittelt. Diese betreffen beispielsweise die Drehzahlstellbarkeit, das Regelverhalten, das dynamische Verhalten unter Berücksichtigung der Ausgleichsvorgänge und den Einfluss der Stromrichterspeisung. Von den mannigfaltigen Sonderausführungen werden nur diejenigen behandelt, die heute noch Bedeutung haben.

Das klar aufgebaute Buch wendet sich an Studenten der Ingenieurwissenschaften, Elektrotechnik und Maschinenbau. Es ist aber auch für den in der Praxis stehenden Ingenieur, der sich über den Einsatz elektrischer Maschinen in Anlagen orientieren will, sehr wertvoll. *A. Meyer*