Research Contribution

Completeness of Logical Functions Realized by Asynchronous Sequential Circuits

HISASHI SATO*, AKIHIRO NOZAKI** and GRANT POGOSYAN***

This paper, concerns completeness problems for logical functions realized by asynchronous circuits that may have feedback loops.

Its first aim is to give mathematical definitions of an asynchronous circuit and of the realization of a logical function by means of an asynchronous circuit. For asynchronous elements, the definitions of circuit construction and initialization are very sensitive: a slight modification may have a considerable influence on the completeness. Several types of completeness are then formulated for a set of logical functions (LF-, GS-, GR-, and NS-completeness).

The second aim is to give a completeness criterion for each tyupe of completeness. This aim is reatized for LF, GS- and GR-completeness. A completeness criterion for NS-completeness is given under a strong condition.

1. Introduction

Digital computers are constructed from basic elements called **logical elements** (or **switching elements**), each of which performs an operation represented by a logical function f: it receives imput signals $x_1, x_2, \ldots, x_n$, and emits an output signal $y=f(x_1, x_2, \ldots, x_n)$ after a certain time-lag. Therefore we are led to ask whether a given set of basic elements is complete, in the sense that it realizes all logical functions. The classical completeness problem deals with sets of logical functions without time-lag. Since a loop of delay-less elements is somewhat contradictory, feedback loops are not allowed. Although the delays of actual logical elements are usually very small in comparison with the duration times of input signals, they are not absolutely zero.

K. Inagaki [1] thowed that use of feedback loops could have a strong influence on the completeness of delayed functions. For instance, after the simple circuit shown in Fig. 1 has received the input signal 0 [in advance], it emits the stable output 0. We can easily show that any loop-free circuit consisting of "AMD" elements cannot realize a constant function. We therefore consider the realization of logical functions by means of sequential circuits. A. Nozaki [2] studied the realization of logical functions by means of sequential circuits with unit delay gates, and introduced S-completeness. He also gave a powerful criterion for. . . .

In this paper, we give mathematical definitions of an asynchronous circuit and of the realization of a logical function by means of an asynchronous circuit. We introduce several types of completeness of a set of logical functions realized by means of asynchronous circuits, and give a completeness criterion for each type of completeness.

*Department of Information and Computer Sciences, Faculty of Engineering, Saitama University.
**Divison of Natural Science, International Christian University.
***Yerevan Polytechic Institute.

2. Logical Functions

Let $M(X, Y)$ be the set of all mappings from a set X to a set Y.

Definition 2.1 *For the set* $\mathbf{B}=\{0, 1\}$, *let*

(i) $\Omega_n=M(\mathbf{B}^n, \mathbf{B})$, *where n is a positive integer,*

(ii) $\Omega=\bigcup_{n=1}^{\infty} \Omega_n$.

We call an element of Ω a logical function.

We represent one-variable logical functions in the following way:

$$I(x) = x$$
$$NOT(x)=\bar{x}=1-x$$
$$C_0(x) = 0$$
$$C_1(x) = 1$$

Let us define some special classes of logical functions. We introduce an order relation into the set B by letting $0<1$.

Let

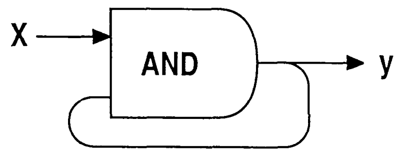

Fig. 1 Realization of 0 by AND.

$$\alpha\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_n)$$

and

$$\beta=(\beta_1, \beta_2, \ldots, \beta_n)$$

be binary sequences. We say that $\alpha\alpha$ is *not greater than* $\beta(\alpha\leq\beta)$ if and only if

$$\alpha_i\leq\beta_i$$

for all $i(1\leq i\leq n)$.

We use the following notations.

$M_0=\{f\in\Omega|f(0, 0, \ldots, 0)=0\}$

$M_1=\{f\in\Omega|f(1, 1, \ldots, 1)=1\}$

S =the set of all self-dual functions $=\{f\in\Omega|f(\overline{x_1}, \ldots, \overline{x_n})=\overline{f(x_1, \ldots, x_n)}\}$

M'=the set of all monotone non-decreasing functions

L =the set of all linear functions

$W=\{f\in\Omega|f(0, \ldots, 0)=1, f(1, \ldots, 1)=0\}$

3. Asynchronous Circuit

A **circuit** C is a system of simultaneous equations as follows:

$$(C)\begin{cases} y_1'=f_1(y_1, \ldots, y_m, x_1, \ldots, x_n) \\ \vdots \\ y_m'=f_m(y_1, \ldots, y_m, x_1, \ldots, x_n) \end{cases}$$

where f_i represents the input-output behavior of the i-th element, whose current output is denoted by y_i, and $x_1, \ldots, x_n$ represent the external inputs given from outside the circuit. The variable y_i' represents the expected output of the i-th element. The first output y_1 is taken as the external output of circuit C.

Let F be a set of logical functions. Circuit C is called a circuit **over** F if and only if all component functions f_i are in F.

Definition 3.1 *A circuit is said to be* loop-free *if and only if the following condition is satisfied:*

The i-th function f_i depends only on $y_{i+1}, \ldots, y_m$ and $x_1, \ldots, x_n$ for all i.

A loop-free circuit can be represented as follows:

$$\begin{cases} y_1'=f_1(y_2, \ldots, y_m, x_1, \ldots, x_n) \\ y_2'=f_2(y_3, \ldots, y_m, x_1, \ldots, x_n) \\ \vdots \\ y_m'=f_m(x_1, \ldots, x_n) \end{cases}$$

Definition 3.2 *A circuit is said to be a* restricted sequential circuit (or more *simply*, a restricted circuit) *or* a short-loop circuit *if and only if the following condition is satisfied:*

The output of an element may be connected to its own input terminal, but no other feed-back loops are allowed. (Informally speaking, all feed-back loops are "short.")

A restricted sequential circuit can be represented as follows:

$$\begin{cases} y_1'=f_1(y_1, \ldots, y_m, x_1, \ldots, x_n) \\ y_2'=f_2(y_2, \ldots, y_m, x_1, \ldots, x_n) \\ \vdots \\ y_m'=f_m(y_m, x_1, \ldots, x_n) \end{cases}$$

An m-tuple

$$\mathbf{y}=(y_1, \ldots, y_m)$$

is called the **state** of circuit C. An n-tuple

$$\mathbf{x}=(x_1, \ldots, x_n)$$

is called the **input** of circuit C.

Definition 3.3 *Let*

$$\mathbf{x}=(x_1, \ldots, x_n)$$

be an arbitrary input of circuit C, and

$$\mathbf{y}=(y_1, \ldots, y_m)$$
$$\mathbf{z}=(z_1, \ldots, z_m)$$

be arbitrary states of circuit C. We say that the state of circuit C can be transferred from $\mathbf{y}$ *to* $\mathbf{z}$ *by the input* $\mathbf{x}$, *if and only if*

$$z_i=\begin{cases} y_i \\ or \\ f_i(y_1, \ldots, y_m, x_1, \ldots, x_n) \end{cases}$$

for all $i(1\leq i\leq n)$. This transition of states is denoted as follows:

$$\mathbf{y}\xrightarrow{\mathbf{x}}\mathbf{z}.$$

If

$$z_i=f_i(y_1, \ldots, y_m, x_1, \ldots, x_n)$$

for all $i(1\leq i\leq n)$, then we say that the transition is **synchronous** *and write*

$$\mathbf{z}=\mathbf{y}'_{(\mathbf{x})}.$$

Definition 3.4 *An infinite sequence of state transitions*

$$\mathbf{y}(0)\xrightarrow{\mathbf{x}}\mathbf{y}(1)\xrightarrow{\mathbf{x}}\mathbf{y}(2)\xrightarrow{\mathbf{x}}\ldots\xrightarrow{\mathbf{x}}\mathbf{y}(t)\xrightarrow{\mathbf{x}}\ldots$$

is said to be **admissible** *for an input* $\mathbf{x}$ *if and only if the following condition is satisfied:*

Let t_0 be an non-negative integer and i a positive integer not greater than m. Let c be can element of B. If

$$f_i(\mathbf{y}(t), \mathbf{x})=c$$

for all $t\geq t_0$, then there exists a moment $T\geq t_0$ such that

$$\mathbf{y}_i(t)=c$$

for any $t\geq T$, where $\mathbf{y}_i(t)$ denotes the i-th component of $\mathbf{y}(t)$.

Remark 3.5 *In practice, we can assume the existence of a positive constant k satisfying the following condition:*
If

$$f_i(\mathbf{y}(t), \mathbf{x})=c$$

for $T \le t < T+k$, *then*

$$y_i(T+k)=c.$$

4. Output Stability

Definition 4.1 *A* synchronous sequence *for an input* x *is a sequence of the form:*

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \dots$$

where

$$\mathbf{y}(t+1)=[\mathbf{y}(t)]'_{\mathbf{x}}$$

for all $t \ge 0$.

Remark 4.2 *A synchronous sequence is always admissible.*

Definition 4.3 *An infinite sequence*

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \dots$$

is said to be output stable *if and only if there exists a non-negative integer* t_0 *such that*

$$\mathbf{y}_1(t_0)=\mathbf{y}_1(t_0+1)=\mathbf{y}_1(t_0+2)= \dots .$$

The output $\mathbf{y}_1(t_0)$ *is called a* stable output *of the sequence.*

Definition 4.4 *An infinite sequence*

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \dots$$

is said to be final state stable *if and only if there exists a non-negative integer* t_0 *such that*

$$\mathbf{y}(t_0)=\mathbf{y}(t_0+1)=\mathbf{y}(t_0+2)= \dots .$$

The state $\mathbf{y}(t_0)=(y_1(t_0), \dots, y_m(t_0))$ *is called* the final stable state, *and the output* $y_1(t_0)$ the final stable output.

Definition 4.5 *Let* t_0 *be a non-negative integer. Let*

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \dots$$

be an admissible sequence. If a state $y(t_0)$ *in the sequence satisfies the following simultaneous equation*

$$\begin{cases} y_1 = f_1(y_1, \dots, y_m, x_1, \dots, x_n) \\ y_2 = f_2(y_1, \dots, y_m, x_1, \dots, x_n) \\ \vdots \\ y_m = f_m(y_1, \dots, y_m, x_1, \dots, x_n), \end{cases}$$

then we have

$$\mathbf{y}(t_0)=\mathbf{y}(t_0+1)=\mathbf{y}(t_0+2)= \dots .$$

We call such a sequence terminating, *and the state* $\mathbf{y}(t_0)$ a stable state.

Remark 4.6 *A terminating sequence is always output stable.*

Proposition 4.7 *Every admissible sequence of a loop-free circuit is terminating.*

Proof. If outputs $y_{k+1}, y_{k+2}, \dots, y_m$ become stable, then output y_k also becomes stable. $\mathscr{Q.E.D.}$

5. Realization of Logical Functions by Asynchronous Sequential Circuits

5.1 Realization

Definition 5.1 *Let Y be a non-empty set of states. Circuit C* realizes *a logical function f in* Ω_n *with respect to the set Y if and only if the following conditions are satisfied for any input* $\mathbf{x}=(x_1, \dots, x_n)$.

(i) *Every admissible sequence starting from a state in Y for the input* x *is final state stable, its final stable state is uniquely determined by the input* x, *and its final stable output is always identical with* $f(x_1, \dots, x_n)$.

(ii) *Let* y and z *be states. If*

$$\mathbf{y} \in Y$$

and

$$\mathbf{y} \xrightarrow{\mathbf{x}} \mathbf{z}$$

for some input x, *then state* z *is also in Y.*

Lemma 5.2 *We consider two circuits, C and C', which realize the logical functions h and k with respect to the state sets Y and Y', respectively:*

$$(C)\begin{cases} y'_1 = f_1(y_1, \dots, y_m, x_1, \dots, x_n) \\ \vdots \\ y'_m = f_m(y_1, \dots, y_m, x_1, \dots, x_n), \end{cases}$$

$$(C')\begin{cases} y'_1 = g_1(y_1, \dots, y_r, x_1, \dots, x_s) \\ \vdots \\ y'_r = g_r(y_1, \dots, y_r, x_1, \dots, x_s). \end{cases}$$

Circuit (C'') *realizes the function*

$$h(k(x_1, \dots, x_s), x_{s+2}, \dots, x_{s+n})$$

with respect to the set:

$$Y''=\{(a_1, \dots, a_m, b_1, \dots, b_r) \mid (a_1, \dots, a_m) \in Y \text{ and } (b_1, \dots, b_r) \in Y'\}.$$

$$(C'')\begin{cases} y'_1 = f_1(y_1, \dots, y_m, y_{m+1}, x_{s+2}, \dots, x_{s+n}) \\ \vdots \\ y'_m = f_m(y_1, \dots, y_m, y_{m+1}, x_{s+2}, \dots, x_{s+n}) \\ y'_{m+1} = g_1(y_{m+1}, \dots, y_{m+r}, x_1, \dots, x_s) \\ \vdots \\ y'_{m+r} = g_r(y_{m+1}, \dots, y_{m+r}, x_1, \dots, x_s). \end{cases}$$

Proof. After the output $y_{m+1}(t)$ of the circuit C' in C'' becomes stable, the circuit C in C'' evaluates the value of

$$h(y_{m+1}, x_{s+2}, \ldots, x_{s+n}),$$

which is equal to

$$h(k(x_1, \ldots, x_s), x_{s+2}, \ldots, x_{s+n})$$

$\mathcal{Q.E.D.}$

5.2 Initialization

There are two contrastive initialization assumptions:

1. **General Initialization Assumption**

We can set the circuit in an arbitrary state by some means.

2. **Initialization-by-Input Assumption**

We can change the initial state only by feeding a certain input sequence to the circuit.

We can use a circuit C for evaluating a function h if it realizes the function with respect to some non-empty set Y of states under the general initialization assumption. Let us consider a circuit C that realizes a function h with respect to a set Y of states under the initialization-by-input assumption. In practice, circuit C cannot be used for evaluating the function h unless there is a finite sequence of inputs that can convert all states of the circuit to some states in Y.

6. Various Types of Completeness

6.1 Classical Completeness Problem

Definition 6.1 *Let F be a set of logical functions. We denote by $\tilde{F}$ the set of all functions constructed by compositions over F. The set F is said to be functional complete if and only if $\tilde{F}=\Omega$.*

E. L. Post gave the following powerful criterion:

Theorem 6.2 ([3]) *Let F be a set of logical functions. F is functional complete if and only if it is not contained in any of the following five sets:*

$$M_0, M_1, S, M^I, \text{ and } L.$$

Definition 6.3 *Let F be a set of logical functions.*

(i) *We denote by $\bar{F}$ the set of all functions realized by loop-free circuits over F.*

(ii) *F is said to be* LF-complete *if and only if $\bar{F}=\Omega$.*

(ii) *F is said to be* LF-incomplete *if and only if it is not complete.*

Theorem 6.4 *Let F be a set of logical functions. F is LF-complete if and only if it is* not *contained in any of the following five sets:*

$$M_0, M_1, S, M^I, \text{ and } L.$$

Proof. Obvious from Lemma 5.2 and Theorem 6.2.

6.2 Completeness under the General Initialization Assumption

Definition 6.5 *Let F be a set of logical functions.*

(i) *We denote by $[F]$ the set of functions each of which can be realized by a circuit over F with respect to some non-empty set Y of states.*

(ii) *F is said to be* GS-complete *if and only if $[F]=\Omega$.*

(iii) *F is said to be* GS-incomplete *if and only if it is not GS-complete.*

Proposition 6.6 *Let F and F'* be sets of logical functions.

(i) $F \subseteq \bar{F} \subseteq [F]$

(ii) $F \subseteq F' \implies [F] \subseteq [F']$

(iii) $F' \subseteq [F] \implies [F \cup F'] = [F]$

(iv) $[[F]] = [F]$

This proposition follows immediately from the definitions.

Proposition 6.7 *The set L of linear functions is GS-*in*complete.*

Proof. Let C be a circuit over L. Let

$$\mathbf{y}(0) \xrightarrow{\mathbf{x}} \mathbf{y}(1) \xrightarrow{\mathbf{x}} \mathbf{y}(2) \xrightarrow{\mathbf{x}} \ldots$$

be an admissible sequence of the circuit for an input $\mathbf{x}$. Suppose that $y_i(t)$ is a linear combination of

$$y_1(0), \ldots, y_m(0), x_1, \ldots, x_n$$

for all i $(1 \le i \le m)$. Since

$$y_i(t+1) = \begin{cases} y_i(t) \\ \text{or} \\ f_i(y_1(t), \ldots, y_m(t), x_1, \ldots, x_n) \end{cases}$$

and

$$f_i \in L,$$

$y_i(t+1)$ is a linear combination of

$$y_1(0), \ldots, y_m(0), x_1, \ldots, x_n.$$

By mathematical induction on t, $y_i(t)$ is a linear combination of

$$y_1(0), \ldots, y_m(0), x_1, \ldots, x_n$$

for all non-negative integers t and all positive integers i not greater than n. $\mathcal{Q.E.D.}$

Proposition 6.8 *The set M^I of all monotone non-decreasing functions is GS-incomplete.*

Proof. Let C be a circuit over M^I. Let

$$\mathbf{y}(0) \xrightarrow{\mathbf{u}} \mathbf{y}(1) \xrightarrow{\mathbf{u}} \mathbf{y}(2) \xrightarrow{\mathbf{u}} \ldots$$

by a synchronous sequence of the circuit for an input $\mathbf{u}$, and

$$\mathbf{z}(0) \xrightarrow{\mathbf{v}} \mathbf{z}(1) \xrightarrow{\mathbf{v}} \mathbf{z}(2) \xrightarrow{\mathbf{v}} \ldots$$

the synchronous sequence of the circuit for another input $\mathbf{v}$, starting from the same state $\mathbf{z}(0)=\mathbf{y}(0)$. Suppose that

$$\mathbf{u} \le \mathbf{v}.$$

Then by mathematical induction on t we can show that

$$y_i(t) \le z_i(t),$$

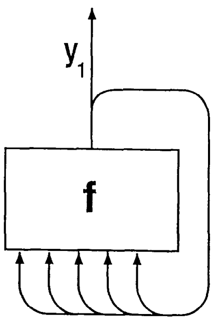

Fig. 2 Circuit C'.

for all non-negative integers t and all positive integers i not greater than n. Therefore the stable output of the circuit is always a monotone non-decreasing function of the input. $\mathcal{Q.E.D.}$

Lemma 6.9 *Suppose that a set F of logical functions is not contained in the set L. Then we can realize both of the constant functions C_0 and C_1 by circuits over F.*

Proof. Let f be a function in $F \backslash L$. Then f is surjective. We write

$$f^*(x)=f(x, x, \dots, x).$$

Let us consider four cases:

(a) If $f^*(x)=I(x)$, then the constant function C_a is realized by the following circuit C' with respect to the state set

$$Y'=\{(a)\}$$

(see Fig. 2):

$$(C')y_1'=f(y_1, y_1, \dots, y_1).$$

(b) If $f^*(x)=NOT(x)$, then the constant function C_a is realized by the following circuit C'' with respect to the state set $Y''=\{(a, 1-a)\}$:

$$(C'')\begin{cases} y_1'=f(y_2, y_2, \dots, y_2) \\ y_2'=f(y_1, y_1, \dots, y_1). \end{cases}$$

(c) If $f^*(x)=C_0(x)$, then we can realize another constant function $C_1(x)$ in the following way. Since f is surjective, there exists a binary sequence

$$\alpha=(\alpha_1, \alpha_2, \dots, \alpha_n)$$

such that

$$f(\alpha_1, \alpha_2, \dots, \alpha_n)=1.$$

The constant function $C_1(x)$ is then realized by the following circuit C''' with respect to the state set $Y'''=\{(1, 0)\}$:

$$(C''')\begin{cases} y_1'=f(y_{2-\alpha_1}, y_{2-\alpha_2}, \dots, y_{2-\alpha_n}) \\ y_2'=f(x, x, \dots, x). \end{cases}$$

(d) If $f^*(x)=C_1(x)$, then we can realize another constant function $C_0(x)$ in a similar way.

In this way we can realize both of the constant functions $C_0(x)$ and $C_1(x)$. $\mathcal{Q.E.D.}$

Theorem 6.10 *Let F be a set of logical functions. F is GS-complete if and only if it is contained in neither M^1 and L.*

Proof. (only-if part) Obvious from Propositions 6.7 and 6.8. (if part) Suppose that F is not contained in L. From Lemma 6.9, we can realize both $C_0(x)$ and $C_1(x)$.

$$C_0(x), C_1(x) \in [F]$$

On the other hand, the set $F \cup \{C_0(x), C_1(x)\}$ is not contained in any of the five sets.

$$M_0, M_1, S, M^1, \text{ and } L.$$

Therefore,

$$\begin{aligned} [F] &= [F \cup \{C_0(x), C_1(x)\}] && (\because \text{Proposition 6.6}) \\ &\supseteq \overline{F \cup \{C_0(x), C_1(x)\}} && (\because \text{Proposition 6.6}) \\ &= \Omega. && (\because \text{Theorem 6.4}) \end{aligned}$$

$$\therefore [F]=\Omega$$

$\mathcal{Q.E.D.}$

6.3 Completeness for Restricted Sequential Circuit under the General Initialization Assumption

Definition 6.11 *Let F be a set of logical functions.*

(i) *We denote by $\lfloor F \rfloor$ the set of all functions that can be realized by a restricted sequential circuit over F.*

(ii) *F is said to be* GR-complete *if and only if* $\lfloor F \rfloor = \Omega$.

(iii) *F is said to be* GR-incomplete *if and only if it is not GR-complete.*

Proposition 6.12 *Let F and F'* be sets of logical functions.

(i) $F \subseteq \bar{F} \subseteq \lfloor F \rfloor \subseteq [F]$

(ii) $F \subseteq F' \implies \lfloor F \rfloor \subseteq \lfloor F' \rfloor$

(iii) $F' \subseteq \lfloor F \rfloor \implies F \cup F' \subseteq \lfloor F \rfloor$

(iv) $\lfloor F \rfloor \subseteq \lfloor \lfloor F \rfloor \rfloor$

This proposition follows immediately from the definitions.

Proposition 6.13 *The sets M^1 and L are GR-incomplete.*

Proof. Obvious from Proposition 6.12 and Theorem 6.10.

We write

$$H=S \cap W.$$

From Theorem 6.10, the set H is GS-complete.

Proposition 6.14 *The set H is GR-incomplete.*

Proof. We consider a one-variable function $h(x)$ realized by a restricted sequential circuit $\mathscr{C}$ over H, with respect to a state set Y.

We will show that $h(x)$ is either $I(x)$ or $NOT(x)$, and cannot be $C_0(x)$ or $C_1(x)$. Assume that the circuit $\mathscr{C}$ is the "minimum" one, having the smallest number of elements among the circuits that realize $h(x)$.

$$(\mathscr{C})\begin{cases} y_1' = f_1(y_1, y_2, \ldots, y_{m-1}, y_m, x) \\ y_2' = f_1(y_2, y_3, \ldots, y_{m-1}, y_m, x) \\ \vdots \\ y_m' = f_m(y_m, x). \end{cases}$$

(Step 1) We shall show that the value of the function $f_m(y_m, x)$ of the m-th element is equal to $NOT(x)$. Two cases are possible:

Case 1 $f_m(0, 0) = f_m(1, 0)$

Since f_m is self-dual, we have

$$f_m(1, 1) = f_m(0, 1).$$

Therefore, the first variable of f_m is dummy, and we have:

$$f_m(y, x) = NOT(x).$$

Case 2 $f_m(1, 0) = 0$

We then have

$$f_m(1, 1) = 0, f_m(0, 1) = 1.$$

Obviously, the output f_m is "unstable" for any input, 0 or 1. This is a contradiction.

(Step 2) Let us examine the function f_{m-1}. Since f_m is identical with NOT, three cases are possible:

Case 1 $f_{m-1}(0, 1, 0) = f_{m-1}(1, 1, 0) = a$

We then have:

$$f_{m-1}(1, 0, 1) = f_{m-1}(0, 0, 1) = \bar{a}.$$

Hence the output of y_{m-1} will eventually become a for input 0, and $\bar{a}$ for input 1.

• $a = 0$

The stable output of f_{m-1} is identical with $I(x)$.

• $a = 1$

The stable output of f_{m-1} is identical with $NOT(x)$. Since $NOT(x)$ is given by y_m, the $(m-1)$-th element is redundant, contradicting the assumption.

Case 2 $f_{m-1}(0, 1, 0) = 1, f_{m-1}(1, 1, 0) = 0$

We then have:

$$f_{m-1}(1, 0, 1) = 0, f_{m-1}(0, 0, 1) = 1.$$

In this case, the output of f_{m-1} is "unstable" for any input. This is a contradiction.

Case 3 $f_{m-1}(0, 1, 0) = 0, f_{m-1}(1, 1, 0) = 1$

We can show that it can be assumed without loss of generality that y_{m-1} is identical with $NOT(x)$.

Thus the $(m-1)$-th element is redundant, contradictory the assumption.

Up to now, we have shown that

(i) $f_m(y, x) = NOT(x)$

(ii) $f_{m-1}(y, NOT(x), x) = I(x)$.

(Step 3) If m is greater than 2, then we can replace the variable y_{m-1} by x, against the minimality assumption of the circuit $\mathscr{C}$. Thus the number of elements is not greater than two. Therefore,

$$h(x) = \begin{cases} NOT(x) & \text{if } m = 1 \\ I(x) & \text{if } m = 2 \end{cases}$$

This completes the proof of the proposition.

Lemma 6.15 *Suppose that F is not contained in L and H. Then we can realize both of the constant functions C_0 and C_1 by circuits over F.*

Proof. (Step 1) Let f be a function in $\Omega \backslash H$. Suppose that f is a non-constant function. Then f is a surjection. We specify that

$$f(x) = f(x, x, \ldots, x).$$

If $f(x)$ is the constant function $C_a(x)$, we can realize the constant function $C_a(x)$. Therefore we can assume that f is a non-constant function.

Case 1 $f(x) = I(x)$

By the same argument as in the proof of Lemma 6.9, we can realize the constant functions C_0 and C_1.

Case 2 $f(x) = NOT(x)$

We then have:

$$f(0, 0, \ldots, 0) = 1, f(1, 1, \ldots, 1) = 0.$$

Since $f \in\!\!/ H$, f is not in S. There then exists a binary sequence

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$$

such that

$$f(\alpha_1, \alpha_2, \ldots, \alpha_n) = f(\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_n) = a.$$

We specify that

$$f(x_1, x_2) = f(x_{2-\alpha_1}, x_{2-\alpha_2}, \ldots, x_{2-\alpha_n}).$$

The constant function $C_a(x)$ is realized by the following $\mathscr{C}'$ with respect to the states set $Y = \{(a, 1), (a, 0)\}$:

$$(\mathscr{C}')\begin{cases} y_1' = f(y_2, x) \\ y_2' = f(x). \end{cases}$$

(Step 2) Let g be a function in $F \backslash L$. Then g is a surjection. We specify that

$$g^*(x) = g(x, x, \ldots, x).$$

Case 1 $g^*(x) = I(x)$

By same argument as in the prrof of Lemma 6.9, we can realize the constant function $C_0(x)$ and $C_1(x)$.

Case 2 $g^*(x) = NOT(x)$

In Step 1, we have shown that a constant function $C_a(x)$ is realized by the circuit over F. Therefore we can realize another constant function $C_{\bar{a}}(x)$.

Case 3 $g^*(x) = C_0(x)$ or $C_1(x)$

By same argument as in the prrof of Lemma 6.9, we can realize the constant functions $C_0(x)$ and $C_1(x)$.

In this way, we can realize both of the constant functions $C_0(x)$ and $C_1(x)$. $\mathscr{Q.E.D.}$

Theorem 6.16 *Let F be a set of logical functions. F is GR-complete if and only if it is not contained in any of the three sets:*

$$M^I, L, \text{ and } H.$$

Proof. (only-if part) Obvious from Propositions 6.13 and 6.14. (if part) Suppose that F is not contained in L and H. From Lemma 6.15, we can realize both $C_0(x)$

and $C_1(x)$:

$$C_0(x),\ C_1(x) \in \lfloor F \rfloor$$

On the other hand, the set $F \cup \{C_0(x), C_1(x)\}$ is not contained in any of the five sets:

$$M_0, M_1, S, M^I, \text{ and } L.$$

Therefore,

$$\begin{aligned} \lfloor F \rfloor &\supseteq \overline{F \cup \{C_0(x), C_1(x)\}} && (\because \text{Proposition 6.12}) \\ &= \Omega. && (\because \text{Theorem 6.4}) \end{aligned}$$

$\mathscr{Q.E.D}$.

6.4 Completeness under the Initialization-by-Input Assumption

Definition 6.17 *A circuit C is said to be* terminating *if and only if (1) every admissible sequence of the circuit C for every input starting from any state is always terminating, and (2) its stable state is uniquely determined by the initial state and the input.*

Definition 6.18 *Let C be a terminating circuit. We denote by*

$$C(\mathbf{y}, \mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3), \ldots, \mathbf{a}(t))$$

the final output of the circuit for the input sequence

$$\mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3), \ldots, \mathbf{a}(t),$$

which is given to the circuit in the following way.

(i) *The first input* $\mathbf{a}(1)$ *is given to the circuit that is in the state* $\mathbf{y}$.

(ii) *The i-th input* $\mathbf{a}(i)$ *is fed to the circuit after the circuit has reached the stable state for the input* $\mathbf{a}(i-1)$. *The final output means the first component of the stable state for the final input* $\mathbf{a}(t)$.

Definition 6.19 *Let C be a circuit. The circuit C* evaluates *a logical function h by an* initial sequence

$$\mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1)$$

if and only if the following conditions are satisfied:

(i) *The circuit C is terminating.*

(ii) *For any input sequence*

$$\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(t)$$

and any state $\mathbf{y}$ *of the circuit C,*

$$h(\mathbf{x}(t)) = C(\mathbf{y}, \mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1), \mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(t)).$$

Lemma 6.20 *Let C be a circuit. if the circuit C evaluates a logical function h by means of an initial sequence*

$$\mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1),$$

then

$$h(\mathbf{x}(t)) = C(\mathbf{y}, \mathbf{x}(-T), \mathbf{x}(-T+1), \ldots, \mathbf{x}(-d-1), \mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1), \mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(t)).$$

for any state $\mathbf{y}$ *and any input sequence*

$$\mathbf{x}(-T), \mathbf{x}(-T+1), \ldots, \mathbf{x}(-d-1)$$

and

$$\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(t).$$

Proof. Obvious from Definition 6.19.

Definition 6.21 *Let F be a set of logical functions.*

(i) *We denote by* $\langle F \rangle$ *the set of all logical functions that are evaluated by a circuit C over F by some initial sequence.*

(ii) *F is said to be* NS-complete *if and only if* $\langle F \rangle = \Omega$.

(iii) *F is said to be* NS-incomplete *if and only if it is not NS-complete.*

Proposition 6.22 *Let F and F′ be sets of logical functions.*

(i) $F \subseteq \bar{F} \subseteq \langle F \rangle \subseteq [F]$

(ii) $F' \subseteq F \implies \langle F' \rangle \subseteq \langle F \rangle$

This follows immediately from the definitions.

Proposition 6.23 *The sets* M^I *and L are NS-incomplete.*

Proof. Obvious from Theorem 6.10 and Proposition 6.22. $\mathscr{Q.E.D}$.

We now have the following theorem.

Theorem 6.24 *Let F be a set of logical functions. Let* $C_0(x), C_1(x) \in F$.

$$F\text{:NS-complete} \iff F\text{:complete}$$

Proof. ($\Longleftarrow$) Obvious from Proposition 6.22.

($\Longrightarrow$) Suppose that F is incomplete. Since $C_0(x)$ and $C_1(x)$ are in F,

$$F \subset M^I \text{ or } F \subset L.$$

From Proposition 6.23, F is NS-incomplete. $\mathscr{Q.E.D}$.

The logical functions $C_0(x)$ and $C_1(x)$ are not in S. For the binary sequence $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, we specify that

$$\bar{\mathbf{y}} = (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_n).$$

Proposition 6.25 *The set S is NS-incomplete.*

Proof. Let C be a circuit over S that evaluates a logical function $h \in \Omega_1$ by means of an initial sequence

$$\mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1).$$

We consider the synchronous sequence of the circuit C for an input x, starting from a state $\mathbf{y}(0)$:

$$\begin{cases} y_1(t+1) = f_1(y_1(t), y_2(t), \ldots, y_m(t), x) \\ \quad\vdots \\ y_m(t+1) = f_m(y_1(t), y_2(t), \ldots, y_m(t), x) \end{cases}$$

for all $t \geq 0$. Let $\mathbf{y}'$ be the stable state for this input x. Since $f_1, f_2, \ldots, f_m$ are in S, we have:

$$\begin{cases} \overline{y_1(t+1)} = f_1(\overline{y_1(t)}, \overline{y_2(t)}, \ldots, \overline{y_m(t)}, \bar{x}) \\ \quad\vdots \\ \overline{y_m(t+1)} = f_m(\overline{y_1(t)}, \overline{y_2(t)}, \ldots, \overline{y_m(t)}, \bar{x}) \end{cases}$$

for all $t \geq 0$. Therefore the synchronous sequence of the circuit C for the input $\bar{x}$, starting from $\overline{\mathbf{y}(0)}$, has stable output $\overline{\mathbf{y}'}$. From Lemma 6.20 and the assumption,

$$\begin{aligned} h(0) &= C(\mathbf{y}(0), \overline{\mathbf{x}(-d)}, \overline{\mathbf{x}(-d+1)}, \ldots, \overline{\mathbf{x}(-1)}, \\ &\qquad \mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1), 0) \\ &= C(\mathbf{y}(0), \overline{\mathbf{x}(-d)}, \overline{\mathbf{x}(-d+1)}, \ldots, \overline{\mathbf{x}(-1)}, \\ &\qquad \mathbf{x}(-d), \mathbf{x}(-d+1), \ldots, \mathbf{x}(-1), \bar{1}). \end{aligned}$$

On the other hand,

$$\begin{aligned} h(1) &= C(\overline{\mathbf{y}(0)}, x(-d)\mathbf{x}(-d+1), \ldots, \mathbf{x}(-1), \\ &\qquad \overline{\mathbf{x}(-d)}, \overline{\mathbf{x}(-d+1)}, \ldots, \overline{\mathbf{x}(-1)}, 1) \\ &= \overline{h(0)}. \end{aligned}$$

Therefore the constant function cannot be evaluated by means of circuits over S. The set S is hence NS-incomplete. $\mathscr{Q.E.D.}$

7. Conclusion

We formulated the notion of completeness for asynchronous circuits, and discussed some variations. We gave completeness criteria for LF-completeness, GS-completeness, and GR-completeness, and gave sufficient conditions for a circuit to be NS-complete.

It is highly desirable to give a general NS-completeness criterion.

References

1. Inagaki, K., *t_6-completeness of Delayed Gates* (in Japanese), *Trans. IECE Japan*, 63-D (November 1980), 827-834.
2. Nozaki, A., *Completeness of Logical Gates Based on Sequential Circuits* (in Japanese), *Trans. IECE Japan*, J65-D (February 1982), 171-178.
3. Post, E. L. *The Two-valued Iterative Systems of Mathematical Logic*, Princeton Annals of Math. Studies, 5, Princeton University Press, Princeton, 1941.

(Received November 13, 1989; revised April 25, 1990)