Digital Sequential Logic Systems without Feedback

Chun-Myoung Park

Division of Electrical Electronic & Information Engineering, Computer Engineering Major,
Chungju National University, Korea,
123 Iryumyun Gumdanri, Chungju City, 380-702, Chungbuk, Korea
Tel. +82-43-841-5346, Fax.: +82-43-841-5340
e-mail : cmpark@gukwon.chungju.ac.kr

Abstract: The digital logic systems(DLS) is classified into digital combinational logic systems(CDLS) and digital sequential logic systems(SDLS).

This paper presents a method of constructing the dsgital sequential logic systems without feedback.

Firstly we assign all elements in Finite Fields to P-valued digit codes using mathematical properties of Finine Fields.

Also, we discuss the operarional properties of the building block T-gate that is used to realizing digital sequential logic systems over Finite Fields.

Then we realize the digital sequential logic systems without feedback.

This digital sequential logic systems without feedback is constructed ny following steps.

Firstly, we assign the states in the state-transition diagram to state P-valued digit dode, then we obtain the state function and predecessor table that is explaining the relationship between present state and previous states.

Next, we obtained the next-state function and predecessor table.

Finally, we realize the circuit using T-gate and decoder.

1. Introduction

Recently the research for more advanced constructing methodology of the sequential logic systems[1-2] is researched by many researcher.

A.D.Singh et al[3] proposed the method of constructing the sequential logic systems based on the universal structure.

And V.P.Srini et al[4] proposed the method of constructing the sequential logic systems using cellular array after constructing the U gate based on + gate and • gate.

Also, T.L.Van et al[5] and R.P.Voith et al[6] propose the universal module ULM-m(Universal Logic Module-m) and applied to it to sequential logic systems.

This paper construction is as following.

Section2 describe the important mathematical properties of Finite Fields and assignment method that all elements over Finite Fields to P-valued digit codes.

Section3 discuss the building block that is used to in this paper and describe the constructing digital sequential logic systems without feedback and circuit realization.

Section4 illustrate the proposed method using an example and compare result with another method.

Finally, section5 summarize the constructing methods of the proposed method and we prospect the future research.

2. Digit code assignment over Finite Fields

This section summarize the important mathematical properties of finite fields and propose the method for assignment elements to digit codes.

2.1 The important mathematical properties of finite fields

Finite Fields(or Galois Fields) is defined by any prime number P and integer m, it is named by $GF(P^m)$. In generally finite fields is organized by 5-tuple $(S,+,\bullet,0,1)$ where S is a set of elements, + and $\bullet$ are binary operation over S, 0 and 1 are each identity element for addition and multiplication arithmetic operation. Also, finite fields is classified into ground fields GF(P) and extension fields $GF(P^m)$. The number of elements over ground fields GF(P), P is the prime number more than 1, are 0,1,2,......,P-1.

The important mathematical properties over finite fields are as following.

[MP 1] Commutative law
(1)$a+b=b+a$ (2)$a\bullet b=b\bullet a$ $(a, b\in GF(P^m))$
[MP 2] Associative law
(1)$a+(b+c)=(a+b)+c$ (2)$a\bullet(b\bullet c)=(a\bullet b)\bullet c$
$(a, b, c\in GF(P^m))$
[MP 3] Distribute law
$a\bullet(b+c)=a\bullet b+a\bullet c$ $(a, b, c\in GF(P^m))$
[MP 4] Zero element 0 exist.
$a+0=0+a=a$ $(a\in GF(P^m))$
[MP 5] Unit element 1 exist.
$a\bullet 1=1\bullet a=a$ $(a\in GF(P^m))$
[MP 6] Inverse element exist.
(1) additive inverse element -a exist.
$a+(-a)=0$
(2) multiplicative inverse element a^{-1} exist.
$a\bullet(a^{-1})=1$ $(a, a^{-1}\in GF(P^m))$
[MP 7] $0\bullet a=a\bullet 0=0$ $(a\in GF(P^m))$
[MP 8] for $\alpha\in GF(P^m)$, $\alpha^{\psi}=\alpha$ and $\alpha^{\psi-1}=1$ $(\psi=p^m)$ in case of $\alpha\neq 1$.
[MP 9] for α, $\beta\in GF(P^m)$ and arbitrary integer m, $(\alpha\pm\beta)^{\mu}=\alpha^{\mu}\pm\beta^{\mu}$ $(\mu=p^m)$
[MP 10] for $\alpha\in GF(P^m)$, $\alpha^{i}\bullet\alpha^{j}=\alpha^{i+j(\mathrm{mod}\ \psi-1)}$ $(\psi=p^m)$

We cited the refernce for the other useful properties.[7-9]

2.2 Digit code assignment

The elements in GF(P^m) are represented by $F(\alpha)=\sum_{i=0}^{m-1} a_i\alpha^i$, where α is root, that is, α have coefficient that have element for integer field Z_P that have mod P, where m degree primitive irreducible polynomial.

$$F(X)=X^m + f_{m-1}X^{m-1} + f_{m-2}X^{m-2} + \ldots\ldots + f_1X^1 + f_0, \qquad (2\text{-}1)$$

Where, $a_i \in Z_P(i=0,1,2, \ldots\ldots ,m-1)$ and $f_0 \neq 0$.

We can represent P-valued digit code for all elements of finite fields using expression (2-1).
Where, the number of digit code of necessity for represent all elements over GF(P^m). Also, we denote MSD(Most Significant Digit) for the most degree coefficient and LSD(Least Sinificant Digit) for the least degree coefficient.
On the oter hand, we denote level for coefficient a_i according to the value, the followinf level and number of elements are generated.

Number of elements for 0 level : ${}_mC_0=1$
Number of elements for m level : ${}_mC_m=1$
Number of elements for 1 level : ${}_mC_1$
Number of elements for 2 level : ${}_mC_2$
……
Number of elements for (m-2) level : ${}_mC_{m-2}={}_mC_2$
Number of elements for (m-1) level : ${}_mC_{m-1}={}_mC_1$

There are (m+1) level generation.
Therefore, we obtain the algorithm of assignment all elements over finite fields to digit code.

[Algorithm]
Step 1 : we represent all elements over finite fields as $e_i(I=0,1,2,\ldots\ldots P^m-1)$
Step 2 : We denote all coefficient to $a_{m-1}\ a_{m-2} \ldots\ldots a_1\ a_0$, also we assign a_{m-1} to MSD and a_0 to LSD.
Step 3 : We assign all efficient 0 to e_0
Step 4 : We assign all efficient (P-1) to denote e_ψ ($\psi=P^m-1$).
Step 5 : ${}_mC_1$ elements of level 1 fill with P-value digit code from LSD, and we assign elements according to sequence.
Step 6 : ${}_mC_2$ elements of level 2 fill with P-value digit code from MSD, and we assign elements using combination.
Step 7 : Until (m-1) level we repeat above step 6.

For example, We apply algorthm to GF(3^2) is as following table 2-1.

Table 2-1. Assign GF(3^2) to 3-valued digit code

Step	Elements	Coefficient	Level
2		a_1 a_0	
3	e_0	0 0	0
5, 6, 7	e_1 e_2 e_3 e_6	0 1 0 2 1 0 2 0	1
5, 6, 7	e_4 e_5 e_7	1 1 1 2 2 1	2
5	e_8	2 2	3

3. The construction of sequential logic systems

The output of sequential logic system is defiended by not only present input but also previous input, differently combinational logic stystem. Therefore present input is feedbacked by delay device or memory device and operated at next time as a input with next time input.
In generally, sequential logic systems is represented as following 5-tuple.

$$M=(S, I, Z, \delta\ , \lambda\ ) \qquad (3\text{-}1)$$

Where, S is a state, I is a input, Z is a output, δ is a next-state function and λ is a output function. Also, S,I,Z ∈ GF(P^m).

The δ have a mappimg relationship as following expression (3-2)

$$\delta\ :\ S_t \times I \rightarrow S_{t+1} \qquad (3\text{-}2)$$

where, S_t is a present state and $S_{t+1}$is a next stae.

On the other hand, λ is presented as following expression (3-3) and (3-4) according to Mealy model and Moore model

Mealy model :

$$\lambda\ :\ S_t \times I \rightarrow Z_{t+1} \qquad (3\text{-}3)$$

Moore model :

$$\lambda\ :\ S_t \rightarrow Z_{t+1} \qquad (3\text{-}4)$$

That is to say, the output of Moore model is constructed by only function of present state.

3.1 Building block T-gate

In this section, we discuss the T-gate as a building block. The block diagram of T-gate is as following fig. 3-1.

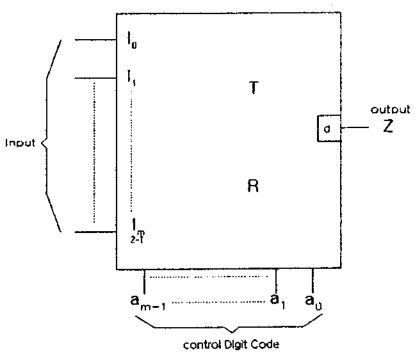

Fig. 3-1. The block diagram of building block is used to constructing the digital sequential logic systems.

Where, I_i is a input and Z is a output.

Also, I_i, Z ∈GF(P^m)(i=0,1,2, ……, P^m-1) and a_j(j=0,1,2, …… , P^m-1) is control digit code.
R is a state digit code V_k(k=0,1,2, …… m-1).

3.2 The constructing digital sequential logic systems without feedback

When constructing sequential logic systems, it is a important to obtain the next-state function. Also, in sequential logic systems, we manipulate the present state and next state. Therefore we represent the next-state function as following expression (3-5) using the time t.

$$S_{t+1}=\delta\ (S_t, I_t) \tag{3-5}$$

Where, S_{t+1}, S_t, $I_t \in e_i \in GF(P^m)(I=0,1,2, \ldots\ldots ,P^m-1)$

We assign state S to state digit code $V_k(k=0,1,2, \ldots\ldots ,m-1)$ using an algorithm that assign elements over $GF(P^m)$ to P-valued digit code . State digit code V_k over $GF(P^m)$ generate V_{m-1}, V_{m-2}, ... , V_1, V_0 and the state equation according V_k is as following (3-6).

$$\begin{aligned} S(V_k)_{t+1} &=(S_0 + S_1 + \ldots\ldots + S_{\zeta -1})_t \bullet\ I_0 \\ &+ (S_0 + S_1 + \ldots\ldots + S_{\zeta -1})_t \bullet\ I_1 \\ &+ \ldots\ldots + (S_0 + S_1 + \ldots\ldots + S_{\zeta -1})_t \bullet\ I_{\zeta -1} \\ &= (\sum_{j=0}^{K-1} S_i)_t \bullet\ I_0 + (\sum_{j=0}^{K-1} S_i)_t \bullet\ I_1 + \ldots\ldots + (\sum_{j=0}^{K-1} S_i)_t \bullet\ I_{\zeta -1} \\ &= \sum_{i=0}^{K-1} (\sum_{i=0}^{K-1} S_i)_t \bullet\ I_j \end{aligned} \tag{3-6}$$

where, $K, \zeta = P^{m-1}$

3.3. The circuit realization of digital sequential logic systems

The block diagram of the circuit of the digital sequentila logic systems is as following fig. 3-2.

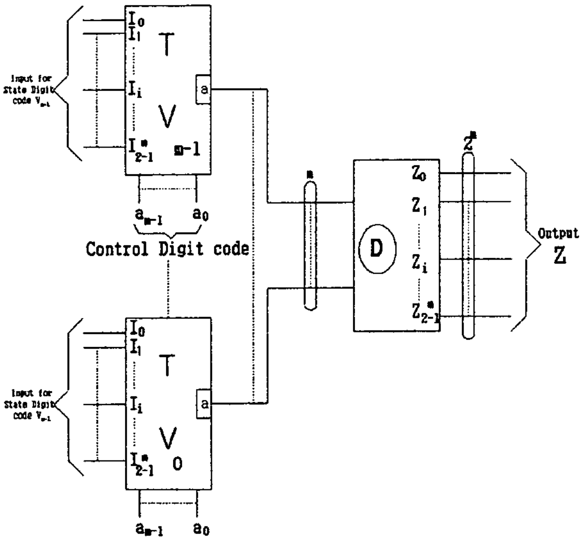

Fig. 3-2. The block diagram of the digital sequential logic systems.

The algorithm of constructing digital sequential logic sysytems without is as following.

[Algorithm]

Step 1 : We obtain the state digit code V_k over the state transition diagram using digit code assignment algorithm

Step 2 : We ontain the state equation according state digit code of step1.

Step 3 : we obtain the predecesspr table from the state transition diagram.

Step 4 : We obtain the next-state transiton function based on state equation and output function from goal state of state transiton diagram.

Step 5 : Finally, we realize the circuit of the digital sequential logic systems without feedback.

4. Application

In this section, we apply the proposed method of constructing the digital sequential logic systems without feedback.

For example, the process of obtain the next-state function from following state-transition diagram fig. 4-1.

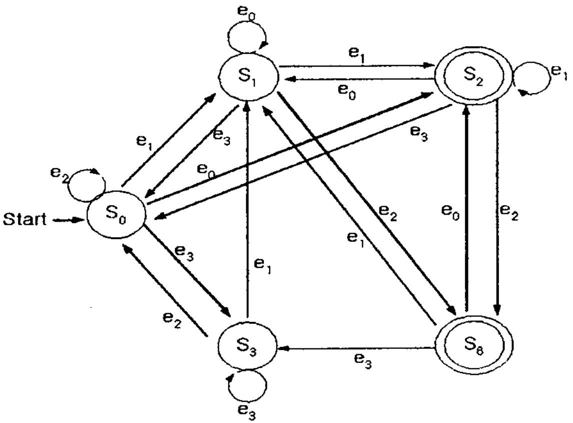

Fig. 4-1. The state transition diagram over $GF(2^3)$.

Step 1 : In fig. 4-1, the used state are S_0, S_1, S_2, S_3 and S_6. we assign state to $S_i(a_1\ a_0)(i=0,1,2,3,4,5,6)$ using digit code assignment algorithm. They $S_0(000)$, $S_1(001)$, $S_2(010)$, $S_3(100)$ and $S_6(010)$. The state represented by state digit code is as following table 4-1.

Table 4-1. An assignment of state in fig. 4-1 to state digit code

Present state	State digit code		
	V_2	V_1	V_0
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	1	0	0
S_6	0	1	1

Step 2 : Next, we obtain the each state equation from table 4-1.

$$V_{2t}=(S_3)_t \tag{4-1}$$
$$V_{1t}=(S_2 + S_6)_t \tag{4-2}$$

$V_{0t}=(S_1 + S_6)_t$ (4-3)

Step 3 : Also, in order to obtain next-state function, we obtain the predecessor table form the Fig. 4-1 , it is following table 4-2,

Table 4-2. the predeccsor table of the fig.4-1.

	Input			
Final state	I_0	I_1	I_2	I_3
S_0	*	*	S_0,S_3	S_1,S_2
S_1	S_0,S_1	S_0,S_3,S_6	*	*
S_2	S_0,S_6	S_1,S_2	*	*
S_3	*	*	*	S_0,S_3,S_6
S_6	*	*	S_1,S_2	*
	Previous state			

Step 4 : we obtain next-state function (4-4) and (4-5) from predecessor table and expression (4-6) and (4-7).

$$S(V_2)_{t+1} = (S_0 + S_3 + S_6)_t \bullet I_3 \quad (4\text{-}5)$$

$$S(V_1)_{t+1} = (S_0 + S_6)_t \bullet I_0 + (S_1 + S_2)_t \bullet I_1 + (S_1 + S_2)_t \bullet I_2 \quad (4\text{-}6)$$

$$S(V_0)_{t+1} = (S_1 + S_2)_t \bullet I_0 + (S_0 + S_3 + S_6)_t \bullet I_1 + (S_1 + S_2)_t \bullet I_2 \quad (4\text{-}7)$$

The output Z is as following.

$Z_t = (S_1 + S_2)_t$ (4-8)

Step 5 : The circuit realization of the digital sequential logic systems for the fig. 4-2 and the comparison table is as following table 4-3.

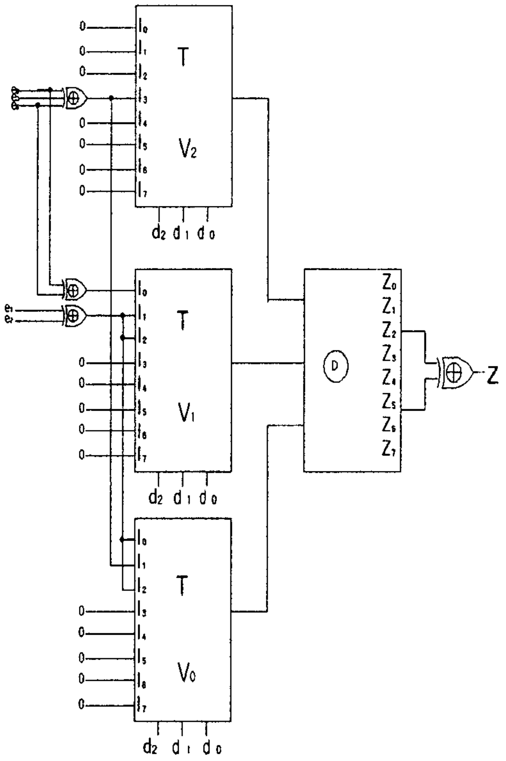

Fig. 4-3. The circuit realization of the digital sequential logic systems without feedback.

Table 4-3. The comparison table.

W.R English[10]	AND 3 OR 1 Delay 1
The proposed digital sequential logic systems without feedback	T-gate 2

5. Conclusion

In this paper, we proposed the method of constructing the digital sequential logic systems without feedback.

The charteristics of the proposed method is as following. Firstly we assign all elements in Finite Fields to P-valued digit codes using mathematical properties of Finine Fields. Also, we discuss the operarional properties of the building block T-gate that is used to realizing digital sequential logic systems over Finite Fields.

This digital sequential logic systems without feedback is constructed by the following steps. Firstly, we assign the states in the state-transition diagram to state P-valued digit dode, then we obtain the state function and predecessor table that is explaining the reletionship between present state and previous states. Next, we obtained the next-state function state function and predecessor table. Finally, we realize the circuit using T-gate and decoder.

References

[1] M.davio, Jean-Pierre, Deschamps and A.Thayse, *Discrete and Switching Functions*, McGraw-Hill international Book company, 1978.

[2] D.Green, *Modern Logic Design*, Electronic Systems Engineering Series, 1986.

[3] A.D.Singh, J.R.Armstrong and F.G.Gray,"Combinational and sequential multivalued logic design using universal iterative tree structures", Int. Symp. on MVL, pp.182-189, 1979.

[4] V.P.Srini,"Iterative realization of multivalued logic systems", Int. Symp. on MVL, pp.188-194, 1978.

[5] T.L.Van and N.V.Houtte, "Delayed universal. logic module and sequential machine synthesis," IEEE Trans. Compt., pp.853-855. Aug. 1975.

[6] R.P.Voith, "Minimum universal logic module sequential circuits with dscoders," IEEE Trans. Compt., vol. C-26, pp.1032-1035, Oct. 1977.

[7] E.Artin, *Galois Theory*, NAPCO Graphics arts, Inc., Wisconsin. 1971.

[8] R.Lidi and G.Pilz, *Applied abstract algebra*, Spring-Verlag, Inc., N.Y., 1984.

[9] R.J.McEliece, *Finite Fields for Computer Science and Engineers*, Kluer Academic Publishers, 1987.

[10] W.R.English, "Synthesis of finite state algorithm in a Galois field GF(P^m)," IEEE Trans. Compt., vol. C-30, pp.225-229, Mar. 1981.