

컴퓨터에 의한 펄스형 순차회로의 설계

(Computer Aided Synthesis for Pulse Mode Sequential Circuits)

황 희 응
조 동 섭
김 병 철*

서울대학교 전자계산기공학과
"
"

요지

본 논문은 펄스형 순차회로(pulse mode sequential circuit)를 설계하는데 필요한 여러가지 복잡한 단계의 간소화를 목적으로 한, 컴퓨터를 이용한 회로 설계법을 제안하고자 한다. 여기서 제안된 방법에 의하면 여러 종류의 플립-플롭(flip-flop)에 대한 회로의 설계를 반복 시행하고, 또 다출력 함수 최소화(multiple output function minimization) 방법을 적용함으로써 거의 최소에 가까운 비용으로 원하는 회로를 설계할 수 있다.

제안된 회로 설계법의 프로그램은 포트란(FORTRAN)으로 작성되었으며, 이에 의한 실제의 예와 그 결과를 종래의 방법에 의한 것과 비교, 분석했다.

1. 서론

순차회로는 두가지 측면에서 분류해 볼 수 있는데 [1], 그 하나는 동작 형태에 따른 것으로서, 동기식(synchronous)과 비동기식(asynchronous)으로, 나머지 하나는 클럭(clock)의 유무에 따라 클럭 인가형(clocked)과 클럭 비인가형(unclocked)으로 분류할 수 있다. 여기서 펄스형 순차회로 경우는 특정한 클럭 펄스는 없지만 입력신상의 펄스 입력에 의해서 전체의 회로가 제어되는 클럭 비인가형 동기식 순차회로이다. 이런 회로의 특성은 회로의 모든 상태들이 안정된 상태(stable state)이며, 불안정된 상태(unstable state)는 존재하지 않는다는 것이다. 즉, 불안정된 상태에 의한 문제점은 발생하지 않게 된다는 것이다. 그러므로 대부분의 연구 방향은 설계비용을 줄이고 동작속도를 높이기 위한 것이다. 또한 다중 입력변

환(multiple input change)에 대한 제약점을 없애기 위한 여러가지 방법도 발표된 바 있다.

[2][3]

하지만 본 논문에서는 위와 같은 세부적인 문제보다도 설계해 나가는 단계들을 컴퓨터를 사용해서 보다 간소화시키고 또 빨리 처리하는 것을 목적으로 한다. 그리고 필요로 하는 함수들을 최소화할 때 함수들간의 공통성(commonality)을 고려함으로써 논리소자수와 연결선수를 줄여서 실제의 회로 설계에 드는 노력과 비용을 줄일 수 있도록 했다.

2. 펄스형 순차회로의 특성

본 논문에서 다루고자 하는 펄스형 순차회로는 앞서 설명한 바와 같이 클럭 비인가형 동기식 순차회로(unclocked synchronous sequential circuit)으로서 다음과 같이 정의될 수 있다.

정의 1. 클럭 비인가형 순차회로가 아래와 같은 조건을 만족하면, 펄스형으로 동작한다고 한다.

(a) 입력은 펄스로서, 그 길이는 플립-플롭을 trigger하기에 충분해야 한다.

(b) 입력은 한 순간에 오직 하나씩만 발생할 수 있다.

(c) 내부 상태(internal state)의 변화는 입력 펄스에 대한 내부 상태의 변화도 하나씩만 일어날 수 있다.

위와 같은 특성을 갖는 회로의 설계는 아래와 같은 일반적인 동기식(synchronous) 순차회로의 설계법에 따르면 된다.

단계1) 설계비용을 줄이기 위해서 주어진 상태표(state table)의 상태를 줄인다.

단계2) Reduced state table의 각 상태에

대해서 상태지정(state assignment)을 한다.

단계3) 위의 상태 지정에 따라 binary-coded 상태 전이표(state transition table)를 구한다.

단계4) 사용하고자 하는 플립-플롭의 특성에 따라 여기표(excitation table)를 작성한다.

단계5) 최소화된 여기함수(excitation function)와 출력함수(output function)를 구한다.

단계6) 구해진 함수에 따라 회로를 설계한다.

이와 같은 설계법은 정의 1.에 따라 동작할 때만을 고려한 것이다. 그러므로 회로의 동작이 제한된 조건을 만족하지 않을 때, 예로써 2개 이상의 입력이 동시에 발생하는 다중 입력변환이 발생할 때는 이런 오동작을 처리하기 위한 특정한 알고리즘(algorithm)을 사용하는 새로운 설계법에 따라야 한다.

이런 특정한 알고리즘을 사용하는 설계법에 대한 간소화가 아니고, 일반적인 모델에 대해 위의 설계법을 적용할 때, 여기변수 Y_k 와 출력변수 Z_l 에 대한 함수들을 구하기 위한 절차를 간소화하게 된다.

3. 펄스형 순차회로의 자동설계 절차

본 논문의 궁극적인 목적은 입력이 입력변수 x_i 와 상태변수 y_j 의 집합인 (x_i, y_j) 로 주어졌을 때, 메모리 소자(memory element)로 사용되는 플립-플롭의 특성에 맞는 출력 y ;와 외부에 대한 출력 z ;를 발생하는 조합회로를 설계하는 것이다. 그러므로 입력과 원하는 출력에 대한 데이터(data)를 제안된 방법에 먼저 입력해 주어야 하는데, 입력될 데이터를 표현하는 데는 몇 가지 고려해야 할 사항이 있다.

- (x_i, y_j) 가 하나의 십진수로서 입력되어 진다. 그 이유는 (x_i, y_j) 가 하나의 상태를 나타내기 때문이다. 이 때 x_i 가 높은 자리수를, y_j 가 낮은 자리수를 나타낸다.

- y_t 와 Z_l 도 각각 십진수로 입력되어 진다.

- 사용되지 않는 상태에 대한 Y_t 나 Z_l 는 don't care state (십진수 -4)로써 입력되어 진다.

정리 1. 입력 x_i 가 2개 이상의 "1"을 갖는 상태 (x_i, y_j) 에 대한 출력 Y_t, Z_l 는 don't care state에 해당하는 값을 갖게 된다.

증명) 펄스형 순차 회로는 정의 1.(b)에 의해서

2개 이상의 입력이 동시에 발생할 수 없다. 그런데 입력이 2개 이상의 "1"을 갖는 상태에서는 그 순간에 2개 이상의 입력이 발생한다는 것을 의미한다. 이런 경우는 생길 수가 없으므로, 이런 상태들에 대한 출력은 모두 don't care로 처리되어야 한다.

위와 같은 조건을 만족하는 입력을 받으면, 제안된 절차(procedure)내에서는 다음과 같은 단계를 거쳐서, 여기함수 g ;와 출력함수 f ;를 구하게 된다.

단계1) 받아들인 십진수의 입력을 이진수로 바꾼다. 이 때 (x_i, y_j) 에 해당하는 십진수를 x_i 와 y_j 각각에 대한 이진수 값으로 분리해서 바꾸게 된다.

단계2) 어떤 플립-플롭을 사용할 것인지를 결정한다.

단계3) 결정된 플립-플롭에 따라서 여기표(excitation table)를 작성한다.

단계4) 작성된 여기표에 따라 여기함수와 출력함수를 최소화 한다.

단계5) 결과를 프린트(print)한다.

단계6) 모든 종류의 플립-플롭에 대해서 단계 2)에서 부터 단계 5)까지를 반복 수행한다.

위의 단계는 일반적인 설계법의 단계3)에서 단계 5)까지에 해당하는 것으로, 단계4), 5)는 다출력 함수 최소화를 위한 알고리즘(MASK-MULTI) [4]에 의한 단계이고, 단계1), 2), 3)은 이에 대한 입력을 작성하는 단계이다.

이와 같은 단계를 거치면 4가지 플립-플롭(SR, JK, D, T)에 대한 여기함수와 출력함수를 얻을 수 있게 된다. 그러면 이 중에서 설계 비용을 최소화하는 것을 선택해서 실제의 회로를 설계하면 된다. 이와 같은 단계를 거쳐서 설계하는 예를 다음 절에서 살펴보기로 한다.

4. 자동설계법에 따른 실제의 예

본 논문에서 제안된 절차에 따라서 모노레일 셔틀 시스템(monorail shuttle system) [5]을 실제로 설계해 보도록 한다. 그림 1.(a)의 각 상태에 대해 상태 지정을 한 결과로, 그림 1.

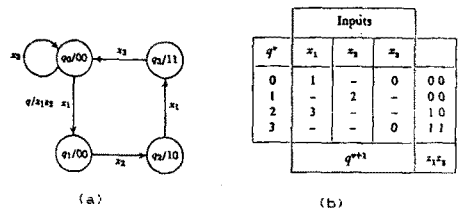


그림 1. 모노레일 제어용 위한 상태 전이도와 전이표

(b)의 상태 천이표(state transition table)를 구할 수 있다.

그림 1.(b)의 상태 천이표를 제안된 설계법에 대한 입력으로 바꾼 형태가 그림 2.와 같고, 이 입력에 대한 출력의 한 예로, SR 플립-플롭에 대한 여기함수, 출력함수가 그림 3.과 같이 된다.

```
State -> 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
N. S. -> 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Output -> 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

그림 2. 제안된 설계법에 대한 입력의 형태

```

.....
* RESULT FOR MULTI-OUTPUT FUNCTION *
.....

I INPUT VARIABLES | CONDITIONALITY FOR SELECTED PARAMOUNT PI
I A B C D E -----
I X 1 X X X      | 1 F1 F2 F3 F4 F5 F6
-----
I X 1 X X X      | 1 0 0 0 0 0 0
I X X 1 X X      | 1 0 1 0 1 0 0
I X X X 1 X      | 1 0 0 0 0 1 0
I X X X X 1      | 1 0 0 0 0 0 1
I X X X 1 X      | 1 0 0 1 0 0 0

```

그림 3. SR 플립-플롭에 대한 결과

위의 제안된 설계법에 의한 결과는 일반적인 설계법에 의한 것과 같음을 알 수 있다. 이 예제에 의한 함수들 간에는 공통성이 없기때문에 일반적인 설계법의 결과와 같아 졌으나, 함수들간에 공통성이 존재하게 되면 일반적인 것보다 더 좋은 결과를 얻을 수 있다. 아래의 표 1.에는 SR 플립-플롭 이외에 JK, D, T에 대한 결과가 나타나 있다.

표 1. 각 플립-플롭에 대한 결과

I	종류	I	여기함수 및 출력함수	I
I		I		I
I		I		I
I	SR	I	S1=X2 R1=X3	I
I		I	S2=Y1X1 R2=X3	I
I		I	Z1=Y1 Z2=Y2	I
I		I		I
I		I		I
I	JK	I	J1=X2 K1=X3	I
I		I	J2=Y1X1 K2=X3	I
I		I	Z1=Y1 Z2=Y2	I
I		I		I
I		I		I
I	D	I	D1=X2+Y1V2+X3Y1V2	I
I		I	D2=X1Y1+Y1Y2+X3Y1Y2	I
I		I	Z1=Y2+Y1V2 Z2=Y2	I
I		I		I
I		I		I
I	T	I	T1=X2+X3Y2	I
I		I	T2=X1Y1+X3Y2	I
I		I	Z1=Y1 Z2=Y2	I
I		I		I

5. 결론

본 논문에 의한 설계법은, 이상에서 살펴 본 바와 같이 일반적인 설계법에 의한 단계를 간소화했고 또한 여러 종류의 플립-플롭에 대해 반복 시행함으로써 가장 적절한 플립-플롭을 결정할 수 있게 해준다. 그리고 함수들간의 공통성을 고려해서 최소화 하기때문에 연결수와 논리소자 수도 줄일 수

있다. 특히 많은 입력과 출력을 갖는 회로일 경우에는 이에 드는 노력과 비용을 크게 절감할 수 있게 된다. 또한 제안된 설계법에 의한 출력에 대해 그래픽 시스템(Graphics system)을 적용하면 CAD시스템으로 발전될 수 있다.

(참고문헌)

- [1] S.C.Lee, Modern Switching Theory and Digital Design. Englewood Cliffs, NJ: Prentice-Hall, 1978
- [2] D.B.Armstrong, "On the Efficient Assignment of Internal Codes to Sequential machines," IRE Trans. on Electronic Computer, vol. EC-11, pp.611-622, Oct. 1962
- [3] D.B.Armstrong, "A Programmed Algorithm for Assigning Internal Codes to Sequential Machine," IRE Trans. on Electronic Computer, vol. EC-11, pp. 466-472, Aug. 1962
- [4] Cho Dong Sub and Hwang Hee Yeung, "A computer algorithm for implementing the Multiple Output Switching Function," KIEE, vol.29, no.10, pp. 678-688, Oct. 1980
- [5] F.J.Hill and G.R.Peterson, Introduction to Switching Theory and Logical Design, 3rd ed. New York Wiley, 1981