

# 비동기 손차 회로의 설계 자동화

( Design Automation of Asynchronous Sequential Circuits )

권 희 용\* (서울대학교)  
 조 동 섭 ( " )  
 황 희 용 ( " )

## 요 약

본 논문은 어떤 비동기적인 제어계가 상태 천이도(transition diagram)로 표현되기만 하면 이 입력으로 부터 직접 비동기 손차회로를 구성할 수 있는 알고리즘을 제시하고 있다. 이로써 비동기 회로를 쉽게 하드웨어로 실현할 수 있도록 하였다.

### 1. 서론

비동기 회로의 설계는 동기회로의 설계에 비해 여러 단계의 복잡한 과정을 거쳐야 하지만, 1969년 Unger 가 상태 변수(State variable)들에 대한 식을 제시하였고 이것을 기초로 1982년 Hollaar 는 비동기 제어 단위의 직접 설계 방법을 제시했다. 이것은 곧 상태 천이도만 있으면 그것을 기초로 바로 회로가 구성될 수 있다는 것을 암시하고 있는 것이다.

이에 본 논문에서는 상태 천이도로 부터 비동기 손차 회로를 직접 설계할 수 있는 컴퓨터 알고리즘을 제시하고 과거의 모든 부자연스런(자연현상의 대부분이 비동기 현상) 동기현상을 쉽게 비동기현상 또는 비동기 제어로 대체할 수 있도록 하고 있다.

제안된 컴퓨터 알고리즘은 비동기 상태 천이도에서 생길수 있는 모든 유형을 다룰수 있도록 Hollaar가 다룬 4가지 기본형을 검토한후 이 기본형들의 복합형과 특수형을 새로이 고쳐한 다음 설계 자동화를 위한 유형 분석에 중점을 두고 있다.

### 2. 비동기형 손차회로의 유형별 분류

Hollaar 는 비동기형 손차 회로를

- (1) 선형
- (2) Mutual Exclusive 형

### (3) Scale-of-two loop 형

의 3가지 기본 유형과 더불어

### (4) 병렬형을 제시하였다.

그런데 그의 제안은 기본형에 대한 것이므로 그 기본형들이 복합될 경우에 대해서는 회로로 직접 실현시킬수 없다는 문제가 있다. 그러므로 직접 실현방법의 일반화를 위해 복합형을 실현시켜야 한다. 설계에 앞서 앞으로 전개에 필요한 용어들을 정의 하기로 한다.

- 단위소자 : 어떤 state 를 나타내는데 필요한 최소의 소자 단위(그림-1)

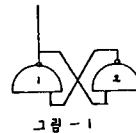


그림-1

- set-input; state를 set 시키는 input 그림-1에서 (1)번 NAND gate의 input(active low) 표시; j state에 대해 SI(j) (line-,...,line-n)
- reset-input; state 를 reset 시키는 input.RI(j)로 표시
- set-line; 어떤 state 의 출력 단자.그림-1의 (1)번 NAND gate 의 출력선
- reset-line; 어떤 state 의 반전 출력단자
- predecessor 와 successor; 그림-2에서 (1)은 (2)의 predecessor (2)는 (1)의 successor 가 된다.

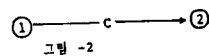


그림-2

- transition input condition; 그림-2에서 c는 (1)에 (2)로의 transition input condition (약해서 input condition)
- FORK state (node);FORK 를 일으키는 state.

그림-3에서 black box.

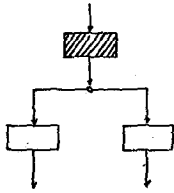


그림-3

• JOIN state; JOIN 되는 state. 그림-4에서 black box.

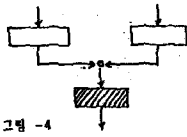


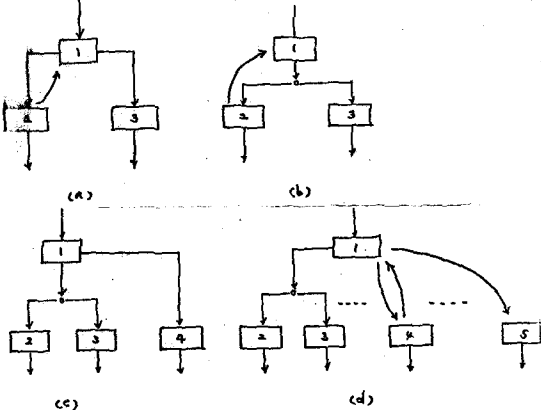
그림-4

• M. E. children state; M.E로 분기되어 연결된 state 들

•  $R(i, j)$ : state  $i$ 와  $j$  와의 relation 이것이 가질수 있는 값으로는

- M.E children state
- JOIN state
- FORK state
- S-O-T relation
- Special relation
- Normal 이 있다.

이상의 용어 정의를 기초로 복합형의 종류를 설계해 본다. 실제로 복합형의 종류는 3가지 (2-3, 3-4, 2-4, 2-3-4)가 존재한다. 이들 각각의 실현 예를 보면 그림-5와 같다.



(c)

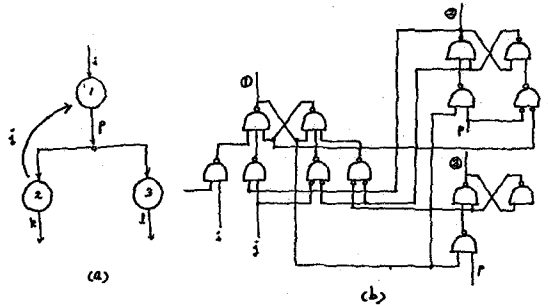
(d)

그림-5 복합형의 종류

그 밖의 다른 경우는 없음을 직관적으로 알수 있다. ( $2^3 - 1 = 7$ ). 그런데 이들은 직접 실현해 보면 이 4가지 복합형중  $b$ 의 경우를 제외하곤 직접 실현에 아무런 모순이 없는 즉 단순히 기본형의 복합에 불과한것을 알수 있다.

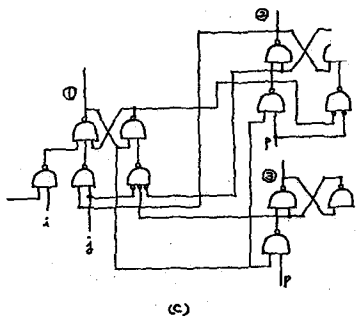
그러나  $b$ 의 경우 단순한 기본형의 복합으로 모순이 생긴다. 다음예를 보자.

그림-6(a)이다. 여기서 (2)에서 (1)로 즉  $j$  input condition 에 의한 transition 은 (1)의 모든 reset input 이 1이 되어야 (1)의 reset 단자가 0로 된다. 그러나 이 경우  $A=0$  이므로 (1)은 set 단자와 reset 단자 모두 1로 불안정 상태로 된다.



(a)

(b)



(c)

그림-6

그러므로 그림-6(c)와 같이 수정할 필요가 있다.

3. 설계 자동화를 위한 계산기 알고리즘

이상으로서 상태 천이도를 회로로 직접 실현시킬수 있게 되었다. 그러나 직접 실현 방법을 알고리즘으로 만들기 위해선 처리 대상에 대해 다음과 같은 사항을 만족하게끔 해야 한다.

첫째, 각 state 는 단위 소자로 나타낼수 있다.

둘째, 각 단위소자는 set-input 과 reset-input 를 갖는다.

이것은 직접 실현 알고리즘은 결국 각 단위 소자

에 대한 set-input 과 reset-input 을 결정하는 데에 그 목적이 있음을 나타낸다.

그림-7은 비동기 순차회로 직접 설계를 위한 알고리즘이다.

```

for j -- 1 to n do
  for i -- 1 to n do
    if S(i,j) ≠ "NULL" then
      if R(i,j)=(M.E.ed children state) AND (JOIN state) then
        SI(j) -- (· predecessor's set line,
          · input condition C,
          · output lines of all another
            M.E.ed children state's input gates)
          /* no increment of input gate */
      else if R(i,j)=(M.E.ed children state) then
        SI(j) -- (· predecessor i's set line,
          · input condition C,
          · output lines of all another
            M.E.ed children state's input gates)
      else if R(i,j)=(JOIN state) then
        SI(j) -- (· predecessor's set line,
          · input condition C)
          /* no increment of input gate */
      else SI(j) -- (· predecessor's set line,
          · input condition C)
      endif
    endif
  repeat
  for i -- 1 to n do
    if S(j,i) ≠ "NULL" then
      if R(j,i)=(Special relation) then
        RI(j) -- (· FORK ed successor's reset line,
          · input condition of scale-of-two
            loop's successor)
          /* no increment */
      else if R(j,i)=(Scale-of-two state) then
        RI(j) -- (· successor's reset line,
          · input condition of that)
      else if R(j,i)=(FORK state) then
        RI(j) -- (· successor's reset line)
          /* no increment */
      else RI(j) -- (· successor's reset line)
      endif
    endif
  end if
  repeat
  repeat

```

그림-7

#### 4. 결론

본 논문에서 제시한 알고리즘에 의해 비동기 제어회로 설계가 직접 설계 가능함을 여러예제를 통해 입증 하였다. 더구나 이 알고리즘의 출력에 의한 비동기 회로는 기본형들의 반복으로 구성되므로써 VLSI 설계에 매우 적합함을 알 수 있다.

그러므로 Petri-Net 와 Data Flow Machine의 하드웨어 실현에 큰 도움이 될수 있으리라 믿는다.

그러나 한가지 주의할 것은 Petri-Net 나 Data Flow Machine 은 모두 비동기현상이긴 하지만 그 표현과 실현에 있어서 약간의 차이를 가지므로 직접적인 응용은 곤란하며 그 밖 에 gate level 에 의한 설계 과정에서 각 신호들간의 지연 시간의 제약이 약간 있으므로 완전한 비동기 신호는 아니라는 점도 남아있다.

DA 를 위해선

HDL

Physical design

layout(positioning, routing)

verification

등이 구비되어야 하지만 이미 verification 과 layout tool 은 존재하므로 새로운 physical design 를 본 논문에서 실현하였다. 앞으로 이 physical design 을 위한 HDL 의 안성이 기대 된다.

(참고 문헌)

- (1) F M. HILL and G.R. PETERSON, 'introduction to Switching Theory and Logical Design', 2-nd edition.
- (2) L.A. HOLLAR, 'Direct Implementation of Asynchronous Control Units', IEEE Transactions on Computers, DEC. 1982.
- (3) J.L. PETERSON, 'Pdtri Nets', computing Surveys, September, 1977.
- (4) G.J. MYERS 'ADVANCES in Computer Architecture', p463-p496, 2-nd edition.
- (5) M.E. DANIEL and C.W. GWYN, 'CAD Systems for IC Design', IEEE Transactions on Computer-Aided Design of IC and Systems, January, 1982.