1. 12주차 예비보고서 20141196 김성희

**Counter**

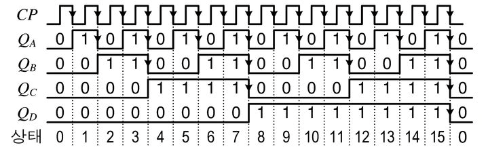
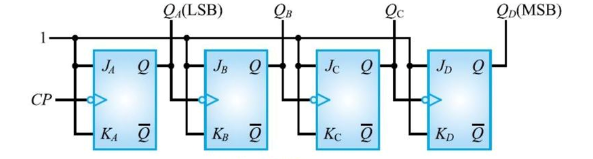
**카운터는 clock의 pulse edge에 따라 여러 개의 플립플롭에 의해 2진수의 숫자가 상승하는 상승 카운터, 숫자가 감소하는 하향 카운터가 있다. 크게 비동기적으로 동작하는 방식과 동기적으로 동작하는 방식으로 나눈다.**

**-비동기식 카운터**

**비동기식 카운터는 각 플립플롭 마다 존재하는 clock 입력 중 첫 플립플롭에만 실제 clock pulse를 입력한다. 그 이후의 플립플롭의 clock 입력은 이전 플립플롭의 출력 값을 이용한다. 보통 JK플립플롭이나 T플립플롭으로 설계가 되는데 이는 토글 기능을 사용하기 위함이다. 우선 상향 카운터에 대한 표를 먼저 살펴보자.  
<상향 카운터>**

**(Q d~a는 각각 4비트짜리 2진수의 각 자리수이며 Qd는 MSB, Qa는 LSB를 의미한다.)**

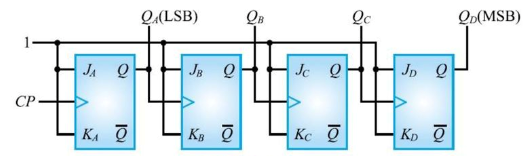
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **클럭펄스** | **Qd** | **Qc** | **Qb** | **Qa** | **10진수** |
| **1** | **0** | **0** | **0** | **0** | **0** |
| **2** | **0** | **0** | **0** | **1** | **1** |
| **3** | **0** | **0** | **1** | **0** | **2** |
| **4** | **0** | **0** | **1** | **1** | **3** |
| **5** | **0** | **1** | **0** | **0** | **4** |
| **6** | **0** | **1** | **0** | **1** | **5** |
| **7** | **0** | **1** | **1** | **0** | **6** |
| **8** | **0** | **1** | **1** | **1** | **7** |
| **9** | **1** | **0** | **0** | **0** | **8** |
| **10** | **1** | **0** | **0** | **1** | **9** |
| **11** | **1** | **0** | **1** | **0** | **10** |
| **12** | **1** | **0** | **1** | **1** | **11** |
| **13** | **1** | **1** | **0** | **0** | **12** |
| **14** | **1** | **1** | **0** | **1** | **13** |
| **15** | **1** | **1** | **1** | **0** | **14** |
| **16** | **1** | **1** | **1** | **1** | **15** |

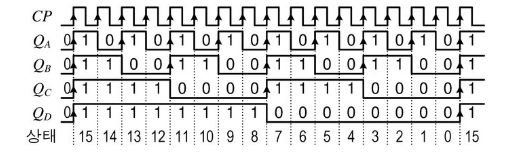
**<표1> 상향 카운터 진리표  
위 표에 대해서 JK플립플롭을 이용해서 비동기식 카운터를 회로로 표현하면 아래 그림 <그림1>과 같다.  
<그림1> JK플립플롭을 이용한 상향 카운터 회로도 (falling edge trigger)**

**<그림2> clock diagram**

**JK플립플롭을 이용하는 이유는 다음과 같다. <표1>을 보면 Qa가 계속 01010101…로 토글 되는 것을 볼 수 있다. <그림1>에서 falling edge마다 JA와 KA에 1을 넣어 줌으로써 이를 구현하였다. 또한 <그림2>를 보면 Qb는 Qa가 1->0일 때, Qc는 Qb가 1->0일 때, Qd는 Qc가 1->0일 때 값이 바뀌는 것을 볼 수 있는데 이는 2진수 연산에서 자리수가 증가할 때는 항상 이전 자리수가 1에서 0으로 바뀌어야하는 부분을 캐치해서 구현한 것이다. 따라서 상향 카운터는 보통 falling edge를 이용해서 만든다.**

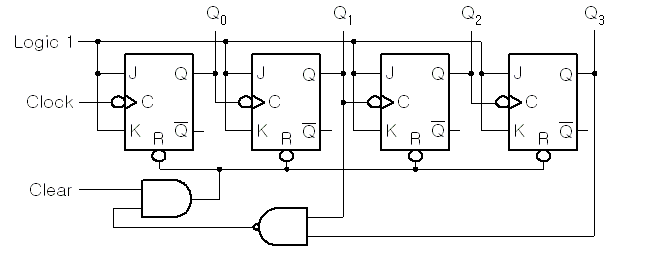
**<하향 카운터>**

**하향 카운터는 1111->1110->1101과 같은 형식으로 줄어든다. 즉 이전 자리수가 0->1로 바뀔 때 현 자리수가 0에서 1로 바뀐다. 마찬가지로 토글 기능을 써야하므로 JK플립플롭이나 T플립플롭을 쓰며 이번에는 0->1로 바뀔 때 다음 자리수가 변경되므로 falling edge대신 rising edge를 쓴다. 회로도와 clock diagram은 아래 그림3, 4를 참조하자.  
<그림3> 하향 카운터 회로도**

**  
<그림4> 하향 카운터 clock diagram**

**<십진 카운터>**

**Decade 카운터는 십진수를 표현하기 위한 카운터를 의미한다. 이를 여러 개 연결해서 여러 자리수의 카운터를 응용하여 만들 수 있다.**

**  
<그림5> Decade카운터**

**0000-> … -> 1000 -> 1001 -> 1010이 되는 순간 clear값으로 1을 보내어 모든 JK플립플롭의 값을 초기화하여 Q0~3모두 0으로 만들어준다. 즉 이를 십진수로 표현해보면 0-> … -> 8 -> 9 -> 0이 된다.**

**-동기식 카운터**

**비동기식 카운터의 단점인 전파지연을 해결하기 위해서 나온 것이 동기식 카운터다. Clock pulse를 첫 플립플롭에 주고 그 결과 값을 다음 플롭플롭의 clock 입력으로 쓰면 결과가 나오기까지의 지연시간이 있는데, 이를 전파지연이라 한다. 동기식 카운터를 이용하기 위해서는 finite state machine에 대해서 알아야한다.**

**-finite state machine**

**Finite state machine(이하 FSM)이란 순차적으로 state를 가지고 결과값을 표현하고 다음 state를 저장하는 기계를 말한다. 따라서 래치나 플립플롭이 필요하며 순차적으로 구현하기 위해 보통 gated latch 혹은 플립플롭을 사용한다. FSM의 구체적은 사용은 표2를 살펴보면서 설명하기로 한다.**

|  |  |  |  |
| --- | --- | --- | --- |
| **현재 상태** | **다음 상태** | | **현재 결과** |
| **X=0** | **X=1** |
| **A** | **A** | **B** | **0** |
| **B** | **A** | **C** | **0** |
| **C** | **A** | **D** | **0** |
| **D** | **A** | **D** | **1** |

**<표2> FSM 예시 진리표 (moore model)**

**표2를 해석 하면, A,B,C,D는 state로 입력값이라 할 수 있다. JK플립플롭을 쓴다고 하면 A,B,C,D는 J와K에 들어가는 수를 결정한다. 표2는 상태가 총 4개이므로 이를 이진수로 표현하면 00 01 10 11과 같은 형식으로 표현한다. 따라서 4개의 플립플롭이 필요하다. 아무튼 표를 해석하면 현재상태가 A이고 입력값 X가 0일 때 결과값은 0이 나오고, 다음 상태는 A라는 의미이다. 즉 00 과 0이 주어지면 결과로, 0이 주어지고 state가 00 그대로 유지된다는 의미이다. 만약 x=1이었다면 00 state에서 01 state로 바뀌었을 것이다.**

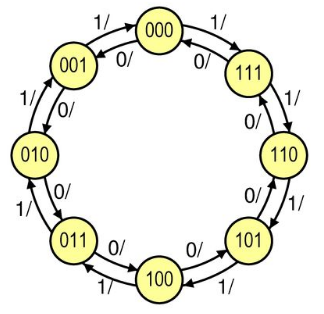
**표2는 moore model이라고 하며 현재 상태와 입력 값으로 다음 상태를 결정하며 출력값은 오로지 현재 상태에만 의존한다는 점이다. 즉 현재 상태가 A,B,C일 때는 결과가 항상 0, D일 때는 1이라는 것이다. 이와 마찬가지로 Mealy model은 다음 상태가 현재 상태와 입력값으로 결정되는 것은 같지만 출력값이 현재 상태와 입력 값에 따라 달라진다. 이를 표로 표시하면 표3과 같은 형식이다.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **현재 상태** | **다음 상태** | | **현재 결과** | |
| **X=0** | **X=1** | **Z=0** | **Z=1** |
| **A** | **A** | **B** | **0** | **0** |
| **B** | **A** | **C** | **0** | **0** |
| **C** | **A** | **D** | **1** | **0** |
| **D** | **A** | **D** | **1** | **1** |

**<표3> FSM 예시 진리표 (Mealy model)**

**이와 같이 moore model과 mealy model을 나눠서 쓰는 이유는 간단하다. Moore model은 입력값이 복잡하지 않아서 mealy model보단 간단하다. 그러나 출력 값을 표현할 때 mealy model이 경우의 수가 더 많으므로 현재 상태의 개수를 줄일 수가 있다. 만약 출력이 입력에 의해 정해지거나 그렇지 않은 경우가 명확하면 각각 mealy model moore model을 고민없이 선택하면 되지만 2가지가 다 가능한 경우라면 상황에 맞게 어떤 것이 적당한지 비교해서 사용하면 될 것이다.**

**<3비트 동기식 상/하향 카운터>**

**여기선 Mealy model을 이용할 것이다. X=0인 경우 상향을, X=1인 경우 하향을 하는 방식이다. 먼저 다음 그림을 살펴 보면 어떤 식으로 작동하는지 알기 쉬울 것이다.**

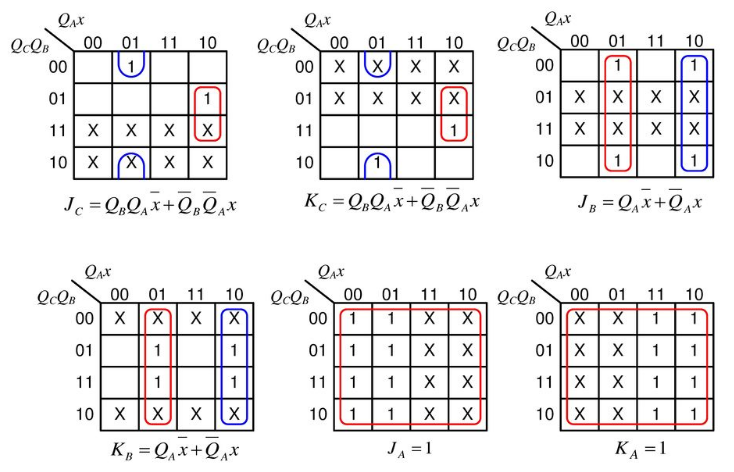
**<그림6> 3비트 동기식 상/하향 카운터**

**이를 mealy model용 진리표로 표현해보기 전에 먼저 JK플립플롭의 입력과 state에 대한 상관 관계에 대해 알아보자.**

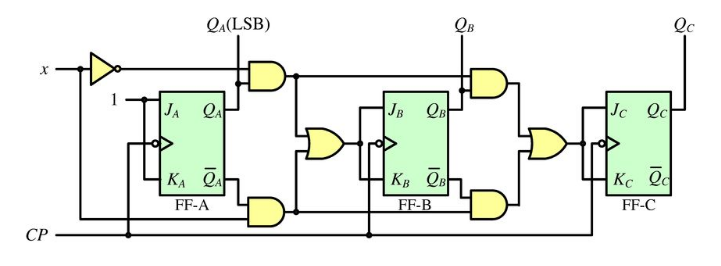
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **현재상태** | **다음상태** | **보존, set, reset, 토글** | **J** | **K** |
| **0** | **0** | **보존, reset** | **0** | **X** |
| **0** | **1** | **토글, set** | **1** | **X** |
| **1** | **0** | **토글, reset** | **X** | **1** |
| **1** | **1** | **보존, set** | **X** | **0** |

**<표4> 현재상태와 다음상태에 따른 JK값 추출 표**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **현재상태** | **입력** | **다음상태** | **플립플롭 입력** | | | | | |
| **QcQbQa** | **X** | **QcQbQa** | **Jc** | **Kc** | **Jb** | **Kb** | **Ja** | **Ka** |
| **000** | **0** | **001** | **0** | **X** | **0** | **X** | **1** | **X** |
| **000** | **1** | **111** | **1** | **X** | **1** | **X** | **1** | **X** |
| **001** | **0** | **010** | **0** | **X** | **1** | **X** | **X** | **1** |
| **001** | **1** | **000** | **0** | **X** | **0** | **X** | **X** | **1** |
| **010** | **0** | **011** | **0** | **X** | **X** | **0** | **1** | **X** |
| **010** | **1** | **001** | **0** | **X** | **X** | **1** | **1** | **X** |
| **011** | **0** | **100** | **1** | **X** | **X** | **1** | **X** | **1** |
| **011** | **1** | **010** | **0** | **X** | **X** | **0** | **X** | **1** |
| **100** | **0** | **101** | **X** | **0** | **0** | **X** | **1** | **X** |
| **100** | **1** | **011** | **X** | **1** | **1** | **X** | **1** | **X** |
| **101** | **0** | **110** | **X** | **0** | **1** | **X** | **X** | **1** |
| **101** | **1** | **100** | **X** | **0** | **0** | **X** | **X** | **1** |
| **110** | **0** | **111** | **X** | **0** | **X** | **0** | **1** | **X** |
| **110** | **1** | **101** | **X** | **0** | **X** | **1** | **1** | **X** |
| **111** | **0** | **000** | **X** | **1** | **X** | **1** | **X** | **1** |
| **111** | **1** | **110** | **X** | **0** | **X** | **0** | **X** | **1** |

**<표5> 동기식 3비트 상/하향 카운터 (mealy model)  
<그림6>과 <표4>를 통해 <표5>를 그릴 수 있다. 예를 들어 000->001인 경우 JcKc는 Qc가0->0을 보고 <표4>에서 0X를 뽑아낼 수 있다. 이제 카르노맵을 통해 Jc~a와 Kc~a에 대한 논리식을 구해보자.**

**<그림7>J와 K의 논리식**

**따라서 이를 JK플립플롭 회로도로 나타내면 아래 그림8과 같다.**