13주차 예비보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20212022 이름: 이예준

**1.**

Flipflop은 1-bit의 정보를 저장하는데, 여러 bit의 데이터를 저장하려면 여러 개의 Flipflop이 필요하다. 이렇게 여러 개의 Flipflop을 직렬로 연결하여 데이터를 저장하는 역할을 하는 것을 Resister라고 한다. 이때, 데이터를 각 Flipflop들을 넘나들면서 이동시키는 회로를 Shift Resister라고 한다. 기본적으로 양방향 FIFO구조이며, Clock Signal에 의해 데이터의 모든 bit이동이 이루어진다. 이런 방식을 통해 현재 계산을 수행 중인 값을 저장하는데 사용되며 메모리 계층 최상위에 위치하여 가장 접근 속도가 빠른 메모리이다. 입력을 직렬로 하냐 병렬로 하냐 또는 출력을 직렬로 하냐 병렬로 하냐 동작 방식에 따라 4가지 종류로 나뉘어진다.

**1. SISO (직렬 입력, 직렬 출력)**

도표, 평면도, 라인, 직사각형이(가) 표시된 사진

자동 생성된 설명

\*SISO Shift Resister 회로도

SISO는 Serial in to Serial out의 약자로 n-bit의 데이터가 직렬로 입력되고, 출력도 직렬로

나온다. Clock Signal이 발생할 때마다 앞에 있는 Flipflop부터 순차적으로 지나가면서

각 Flipflop에 데이터가 저장된다. 따라서 Clock Signal이 발생하지 않을 때는 데이터를

유지시킨다. Clock Signal에 의해 순차적으로 이동하기 때문에 SISO의 주 목적은 지연 요소

역할을 하는 것이다. Clock Signal이 모든 Flipflop에 연결되어 있기 때문에 서로 동기화된다.

텍스트, 도표, 회로이(가) 표시된 사진

자동 생성된 설명

\*SISO Shift Resister 구조도

**2. SIPO (직렬 입력, 병렬 출력)**

도표, 평면도, 라인, 직사각형이(가) 표시된 사진

자동 생성된 설명

\*SIPO Shift Resister 회로도

SIPO는 Serial in to Parallel out의 약자로 n-bit의 데이터가 직렬로 입력되고, 출력은 병렬로

나온다. SISO와 마찬가지로 앞에 있는 Flipflop부터 순차적으로 지나가면서 각 Flipflop에

데이터가 저장되지만 출력이 나오는 것은 현재 각 Flipflop이 저장하고 있는 값들을 모두

병렬로 출력해준다. SIPO는 직렬의 데이터를 병렬 데이터로 변환하는 과정이며 데이터를

선택을 통해 여러 Output으로 Demultiplexing해야 하는 통신에 사용된다. 마찬가지로

Clock Signal이 모든 Flipflop에 연결되어 있기 때문에 서로 동기화된다.

텍스트, 도표, 회로이(가) 표시된 사진

자동 생성된 설명

\*SIPO Shift Resister 구조도

**3. PISO (병렬 입력, 직렬 출력)**

도표, 평면도, 기술 도면, 개략도이(가) 표시된 사진

자동 생성된 설명

\*PISO Shift Resister 회로도

PISO는 Parallel in to Serial out의 약자로 n-bit의 데이터가 병렬로 입력되고, 출력은 직렬로

나온다. 데이터는Multiplexer에 의해 데이터의 각 bit가 모든 Flipflop에 병렬로 저장되며, Clock Signal에 맞춰서 순차적으로 1-bit씩 데이터를 출력으로 내보낸다. PISO는 병렬의 데이터를 직렬 데이터로 변환하는 과정이며, 다른 소스의 데이터를 순차적으로 저장하고 나중에 분석 또는 추가처리를 하기 위해 접근하는 Data Logging에 사용된다. 마찬가지로Clock Signal이 모든 Flipflop에 연결되어 있기 때문에 서로 동기화된다.

텍스트, 도표, 회로이(가) 표시된 사진

자동 생성된 설명

\*PISO Shift Resister 구조도

**4. PIPO (병렬 입력, 병렬 출력)**

도표, 라인, 평면도, 직사각형이(가) 표시된 사진

자동 생성된 설명

\*PIPO Shift Resister 회로도

PISO는 Parallel in to Parallel out의 약자로 n-bit의 데이터가 병렬로 입력되고,

출력은 병렬로 나온다. 데이터의 각 bit는 모든 Flipflop에 병렬로 저장되며, 출력도 병렬로

데이터가 나오는데, Clock Signal에 맞춰서 순차적으로 데이터의 각 bit가 현재 저장된

Flipflop에서 옆에 있는 Flipflop으로 이동한다. 예를 들어. 1101을 입력하면 처음 출력은

1101이지만 Clock Signal이 한번 발생한 뒤에는 0110으로 출력된다. 그 다음은 0011이 된다. 마찬가지로Clock Signal이 모든 Flipflop에 연결되어 있기 때문에 서로 동기화된다.

텍스트, 도표, 회로이(가) 표시된 사진

자동 생성된 설명

\*PIPO Shift Resister 구조도

**2.**

위에서 설명한 Shift Resister는 입력이 들어오면 순차적으로 이동하면서 출력으로 다

나온 뒤 끝나는 구조인데, Ring Counter는 Shift Resister처럼 데이터가 순차적으로 이동하는 건 같지만, 마지막에 있는 Flipflop에서 나온 출력을 다시 맨 앞에 있는 Flipflop의 입력으로 받아 데이터가 계속해서 순환하는 Counter이다.

도표, 평면도, 라인, 기술 도면이(가) 표시된 사진

자동 생성된 설명

\*Ring Counter 회로도

Ring Counter는 두가지 종류가 있는데, Straight Ring Counter와 Twisted Ring Counter가

|  |  |  |  |
| --- | --- | --- | --- |
| Q0 | Q1 | Q2 | Q3 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 |

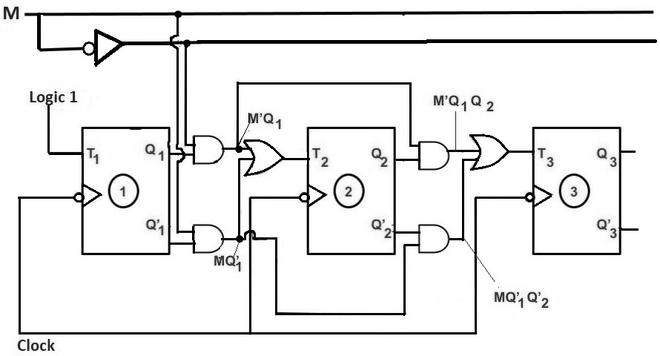
있다. Straight Ring Counter는 위의 회로도와 같이 마지막 출력이 입력으로 다시 순환하는 구조이며 하나의 0 또는 1을 순환시킨다. Twisted Ring Counter는 마지막 Flipflop의 출력의 보수를 첫번째 Flipflop의 입력으로 입력 받는 구조로 1의 흐름이 온 뒤 0의 흐름이 온다.

|  |  |  |  |
| --- | --- | --- | --- |
| Q0 | Q1 | Q2 | Q3 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

\*Straight 진리표 \*Twisted 진리표

**3.**

UP DOWN Counter는 Mode Control Input에 의해 Count값을 증가시키는 방향으로 바꾸거나 감소시키는 방향으로 바꿀 수 있다. 위에서 설명한 Counter와 마찬가지로 Clock Signal이 모든 Flipflop과 연결되어 서로 동기화한다. 또한 Counter 회로에 Mode Control을 위한 논리회로를 추가시킬 필요가 있다.



\*UP DOWN Counter 회로도

|  |  |  |  |
| --- | --- | --- | --- |
| M | Q3 | Q2 | Q1 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |

M값이 up, down을 결정하는데, 만약 M이 0이라면 Counting이 증가하는 방향이 되며, M이 1이라면 Counting이 감소하는 방향이 된다.

|  |  |  |  |
| --- | --- | --- | --- |
| M | Q3 | Q2 | Q1 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

\*UP Counter 진리표 \*Down Counter 진리표

**4.**

위에서 설명한 Counter는 하나의 Clock Signal이 모든 Flipflop과 연결되어 동기화되어 있는데, Ripple Counter는 맨 앞에 있는 Flipflop만 Clock Signal을 받고 나머지 Flipflop은 이전 Flipflop의 출력을 받는다. 따라서 비동기식 Counter라고도 한다.

도표, 평면도, 스케치, 기술 도면이(가) 표시된 사진

자동 생성된 설명

\*Ripple Counter 회로도

첫 번째 Flipflop에 Clock Signal이 들어오면 처음에 저장된 데이터가 0이기 때문에 Q1’은

1이 나오면서 입력으로 들어간다. 또한 Q1’의 1이 두번째 Flipflop의 Clk으로 들어간다.

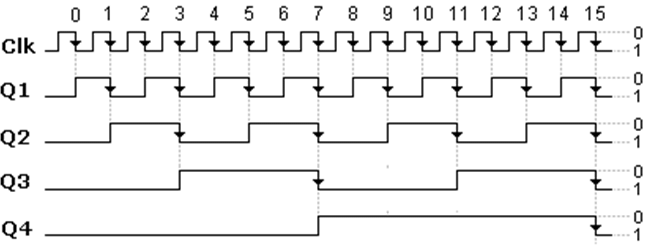
다시 Clock Signal이 첫번째 Flipflop으로 들어가면 전에 저장했던 1을 출력으로 내보내고,

두번째 Clk에 들어왔던 1에 의해 두번째 Flipflop이 작동되면서 Q2’가 1이 나오면서

입력으로 들어간다. 또한 Q2’의 1이 세번째 Flipflop의 Clk으로 들어간다. 이렇게 각

Q값들이 동시에 변하지 않고 일정한 간격을 두면서 변화하는 이 현상을 glitch현상이라고

한다. 아래의 그림은 Clock Signal에 따라 변하는 Q값을 보여주는 Timing Diagram이다.



\*Ripple Counter Timing Diagram

Ripple Counter는 T 또는 D Flipflop으로 간단하게 구현할 수 있다는 장점과 첫번째 Flipflop만 Clock Signal과 연결해도 되어 동작 및 논리회로 구성이 단순하다는 장점이 있다. 하지만 재동기화를 위해 더 많은 Flipflop이 필요하며, 각 Flipflop을 통과할 때마다 Propagation Delay가 발생하기 때문에 bit수가 많아질수록 Propagation Delay가 매우 커진다.

**5.**

동기식 Parallel Carry Counter는 Flipflop들을 병렬로 연결하고, 뒤에 있는 Flipflop은

앞에 있는 Flipflop들이 모두 1을 출력해야 작동한다. Counter가 동기식이기 때문에

Clock Signal은 모든 Flipflop과 연결되어 동시에 신호를 준다.

도표, 평면도, 라인, 기술 도면이(가) 표시된 사진

자동 생성된 설명

\*Parallel Carry Counter 회로도

위의 그림을 보게 되면 T0는 1이 들어오며, Q0가 1일 때 T1에 입력이 들어오고, Q0와 Q1이

1일 때 T2에 입력이 들어오고, 마지막 T3에 입력이 들어올 때는 Q0, Q1, Q2가 모두 1일

때이다. Parallel Carry Counter는 Flipflop의 수나 수를 Counting하는 것이 time period와

무관하기 때문에 Counting Sequencing의 크기가 어떻든 time period는 일정하게 유지된다.

하지만 위의 회로도를 보면 알 수 있듯이 Flipflop의 수가 늘어나면 늘어날수록

AND Gate도 그만큼 많이 필요하게 된다. 따라서 작은 크기의 Counter에는 유리하지만

큰 크기의 Counter에는 그렇지 못하다.