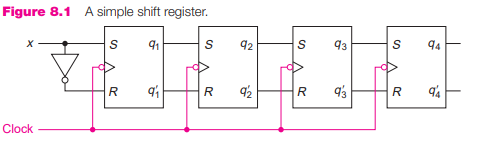
13주차 예비보고서

전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

1. **Shift register에 대해 조사하시오.**

지난 실습에 flip-flop을 직렬로 연결하여 정보를 저장하고 이를 컴퓨터 구조의 RAM의 register에서 사용한다고 조사했다. Register는 flip-flop의 직렬 연결로 flip-flop의 특성을 가지고 있다. 정보를 저장하는 기능을 갖지만 전원이 꺼지면 저장된 정보도 같이 사라지는 휘발성 특성을 갖기에, 현재 process의 정보를 저장하는 RAM에서 사용한다. Shift register는 데이터를 저장하고 이동시키는 회로로 앞서 counter와 비슷한 구조인 flip-flop을 직렬로 연결한 구조를 갖는다. 각 flip-flop의 출력은 그 다음 flip-flop의 입력으로 들어간다. Register의 특징으로는 직렬구조 이외에 병렬 구조도 사용한다는 것이다. 입력과 출력을 모두 직렬 또는 병렬로 원하는 대로 조정할 수 있다. 직렬은 Serial의 약자로 S, 병렬은 Parallel의 약자로 P, 입력과 출력은 I, O로 나타낸다. 따라서 직렬 입력과, 직렬 출력은 SISO와 같이 나타낼 수 있다. 직렬 입력은 각 flip-flop의 출력을 다음 입력으로 넣는 방식이고, 병렬 입력은 각 flip-flop의 출력과 또 하나의 각 flip-flop의 입력을 추가로 설정하여 선택하는 방식이다. 출력 또한 직렬 출력은 모든 flip-flop의 결과를 하나의 출력결과로 사용하는 것이고 병렬 출력은 각 flip-flop의 결과를 각각 출력 값으로 나타내는 것이다. 또한 shift의 말 그대로 데이터를 옮기는 기능을 한다. Shift register는 첫 flip-flop의 결과를 다음 flip-flop의 결과로 옮긴다. 예를 들어 첫 flip-flop의 저장된 정보 Q0가 1이고 다음 flip-flop에 저장된 정보 Q1이 0이라 할 때, 회로가 한번 동작하면 Q0=0, Q1=1이 된다. 가장 간단한 shift register의 구조는 다음과 같다.



이 구조에서 x값에 따라 첫 flip-flop에 정보가 저장된다. 이후 회로가 동작함에 따라 q1값은 q2로, q2값은 q3로 순차적으로 이동한다. 이를 timing으로 보면 다음과 같다.



1. **Ring counter에 대해서 조사하시오.**

Ring counter는 위의 shift register를 ring형태로 연결한 것이다. 마지막 출력을 다시 첫 flip-flop의 input으로 하여 counter의 기능처럼 마지막 출력에 도달 시 0으로 초기화하여 순차적으로 출력 값을 만든다. 이 ring counter의 동작에 대한 예시를 표로 보면 더 쉽게 알 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock** | **Q0** | **Q1** | **Q2** |
| **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **0** |
| **2** | **0** | **0** | **1** |
| **3** | **1** | **0** | **0** |
| **4** | **0** | **1** | **0** |

위의 표처럼 clock이 발생함에 따라 shift register의 역할인 데이터를 다음 flip-flop으로 넘겨준다. 하지만 마지막에 출력 값을 단순하게 나타내는 shift register와 다르게 다시 처음 flip-flop에 넘겨주어 순환하는 counter의 기능을 한다.

1. **UP DOWN counter에 대해서 조사하시오.**

UP DOWN counter는 지난 실습 구현했던 counter들에 기능을 추가한 counter이다. 2-bit binary counter를 예시로 들어보면 지난 실습에서 00부터 11까지 순차적으로 증가하는 counter를 구성했다. 여기에 기능을 추가하여 단순 증가만 시키는 것이 아닌 감소시키는 기능을 추가하여 증가 또는 감소를 우선 선택하고 현재 저장된 정보에서 다음 정보를 나타내는 회로가 UP DOWN counter이다. UP DOWN을 선택하는 변수를 x라 하고 x=1일 때 UP을, x=0일 때 DOWN으로 2-bit UP DOWN counter에 대한 동작 예시를 표로 나타내면 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q0 | Q1 | X | Q0\* | Q1\* |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

위와 같이 이전 상태와 X에 따라 다음 상태 정보가 저장되는 것을 알 수 있다. 2-bit binary counter를 구성할 때, 각 자릿수 bit에 맞게 flip-flop을 구성하고 해당 flip-flop의 Q, 즉 현재 저장 상태를 사용했다. UP DOWN counter를 구성할 때는 Q’도 사용하여 감소시키는 counter를 같이 구성한다. 따라서 2-bit UP DOWN counter도 2개의 flip-flop을 이용하여 이를 직렬로 연결하고 첫 flip-flop의 저장 정보인 Q와 Q’모두 다음 flip-flop의 input으로 사용된다.

1. **Ripple counter에 대해서 조사하시오.**

Ripple counter는 다른 말로 asynchronous counter라고 불린다. 이는 저번 실습에서 조사했던 동기식, 비동기식 counter와 관련이 있다. Counter의 구성 방식 중 clock pulse를 사용함에 있어 각 flip-flop이 모두 함께 작동하도록 하는 것이 동기식 counter였다. 이 동기식 counter는 clock pulse가 각 flip-flop의 input으로 들어가 cp가 발생하면 회로 전체가 함께 작동한다. 따라서 회로 동작에 있어 걸리는 시간이 짧은데 비해 회로 구성이 어렵다는 단점이 있었다. 이에 반해 clock pulse가 가장 처음 flip-flop에 전해지고 처음 flip-flop이 작동한 후 결과값이 다음 flip-flop의 clock pulse처럼 작용하는 회로를 비동기식 counter라고 했다. 이는 각 flip-flop이 전 단계의 flip-flop이 동작하고 나서 동작하는 방식으로 작동된다. 이 비동기식 counter는 flip-flop들이 순차적으로 동작함에 있어 지연시간이 길다는 단점이 있다. 하지만 동기식 counter에 비해 회로 구성이 단순하다는 장점이 있다. 이 비동기식 counter를 다른 말로 ripple counter라고 부른다.

1. **기타 이론.**

Register는 shift register 이외에도 다양한 register가 존재한다. Shift register가 데이터를 옮기는 기능을 한다면 status register는 상태를 기록하는 기능을 하는 register이다. 여기서 상태란, 이전 실습에서 구현했던 adder를 통해 알 수 있다. adder에서 carry가 발생하거나, overflow가 발생하는 상황이 있었다. 또한 carry를 계산할 때, carry look ahead adder를 통해 carry가 발생하는 자리를 빠르게 계산하는 회로도 구성했다. 여기서 carry가 발생하는 상태, 또는 overflow가 발생하는 상태를 저장하는 것이 status register이다. 이 뿐만 아니라, 계산이 완료한 상태, OS에서 interrupt가 발생한 상태 등 다양한 상태를 저장하는 register가 status register이다.