|  |
| --- |
| **조별 과제: 스톱워치** |

|  |  |
| --- | --- |
| 날짜: | 2022. 11. 24 |
| 학번, 이름: | 21700384 송태웅 21700794 홍준형 |

1. **목적**

* 7 segment와 clock counter 등을 이용하여 스톱워치를 구현할 수 있다.
* PS2키보드와 UART 통신을 이용하여 스톱워치를 제어할 수 있다.

1. **과제 내용**

|  |  |
| --- | --- |
| **출력** | 7 segment 6개 |
|  | (7 segment display의 우측에서부터 1/100초, 1/10초, 1초, 10초, 1분 단위로 표시) |
| **상태** | 0: 정지 및 초기화 |
|  | 1: 작동(시간이 증가하는 상태) |
| **입력** | 1) PS2키보드 입력 |
|  | * ENTER 입력: 스톱워치 reset * s 입력:   (초기화 후 누르는 경우) 스톱워치 카운트 시작  (일시정지 후 누르는 경우) 일시정지된 이후부터 카운트 재시작   * t 입력: 스톱워치 일시정지   2) UART 키보드 간접 입력 |
|  | * (위의 내용과 동일) |

1. **Flowchart**

|  |
| --- |
|  |
| **그림 1. 스톱워치 회로도(PS2KEYBOARD)** |

|  |
| --- |
|  |
| **그림 2. 스톱워치 회로도 (UART)** |

1. **코드 및 기능**

* **Top module**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **그림 3. Top module 코드** | | |

Top module은 그림 1의 회로도에 맞게 코드를 구성하였다.

* **Sub module**
  + **stop\_watch\_control**

|  |  |
| --- | --- |
|  |  |
| **그림 4. Sub module 코드 (1)** | |

stop\_watch\_control 모듈은 스톱워치 제어를 위한 제어신호를 생성한다. 키보드 ‘s’를 눌렀다 뗄 경우, 스톱워치 동작을 의미하는 is\_start 제어신호가 켜진다. 키보드 ‘t’를 누르는 경우, 스톱워치 일시정지를 의미하는 is\_stop 제어신호가 켜진다. 마지막으로 키보드 ‘ENTER’를 누르는 경우, 스톱워치 초기화를 의미하는 is\_reset 제어신호가 켜진다.

* + **clock\_counter\_top**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **그림 5. Sub module 코드 (2)** | | |

clock\_counter\_top 모듈은 clock divider의 기능을 하는 clk\_div\_under\_100 모듈과 binarry counter BC6와 BC9로 이루어져 있다. clk\_div\_under\_100은 100MHz인 입력 clock으로 스톱워치의 1/100초(100Hz)를 구현하기 위해 clock의 rising edge마다 증가하는 count변수가 번마다 enable 신호를 켜도록 구현하였다. 이 enable 신호를 1/100초 단위의 시간을 count하는 BC9의 increment입력으로 주었으며, 이를 기준으로 1/10초, 1초, 10초, 1분, 10분 단위의 시간을 count하는 BCD counter를 자동으로 동기화할 수 있다.

* + - **clk\_div\_under\_100**

|  |
| --- |
|  |
| **그림 6. Sub module 코드 (3)** |

* + - **BC6**

|  |
| --- |
|  |
| **그림 7. Sub module 코드 (4)** |

* + - **BC9**

|  |
| --- |
|  |
| **그림 8. Sub module 코드 (5)** |

1. **결과**

* **Testbench**
  + **stop\_watch\_control**

|  |  |
| --- | --- |
|  |  |
| **그림 9. testbench 코드 (1)** | |

|  |
| --- |
|  |
| **그림 10. 시뮬레이션 결과 (1)** |

stop\_watch\_control 모듈의 시뮬레이션을 통해 테스트 입력에 대해 올바른 제어신호가 생성되는지 확인하였다. 모든 값이 초기화 된 상태에서 먼저, 키보드 ‘s’에 해당하는 값(‘h73)과 released bit에 1을 주어 키보드 ‘s’를 눌렀다 뗀 경우의 테스트 입력을 주었을 때, 스톱워치 작동을 의미하는 제어신호인 is\_start bit가 1로 켜진 것을 확인할 수 있다. 다음으로 키보드 ‘t’에 해당하는 값(‘h74)을 준 경우, 스톱워치 일시정지를 의미하는 제어신호인 is\_stop bit가 1로 켜지고, is\_start bit는 0으로 꺼지는 것을 확인할 수 있다. 키보드 ‘ENTER’에 해당하는 값(‘h0D)을 준 경우, 스톱워치 초기화를 의미하는 제어신호인 is\_reset bit가 1로 켜지고, 나머지 제어신호는 0으로 꺼지는 것을 확인할 수 있다. 마지막으로, 키보드 ‘s’에 해당하는 값(‘h73)과 released bit가 0인 경우에는 제어신호의 변화가 없음을 확인할 수 있다.

* + **BC6**

|  |
| --- |
|  |
| **그림 11. testbench 코드 (2)** |

|  |
| --- |
|  |
| **그림 12. 시뮬레이션 결과 (2)** |

위 그림은 4 bit BCD counter에 modulo 5 연산을 적용하여 0부터 5까지 카운트하는 BCD counter에 대한 시뮬레이션 결과이다. increment를 의미하는 inc 신호는 항상 1로 켜진 상태를 가정하고, 시뮬레이션 하였다. 먼저, counter 출력인 Q가 0부터 5까지 증가한 후, TC bit가 1로 켜진다. 이후 다시 Q는 0으로 돌아가고, TC bit도 0으로 꺼지는 것을 확인할 수 있다. 추가적으로 스톱워치 컨트롤러에서 출력되는 is\_reset 신호가 1로 켜진 경우, Q가 0으로 초기화 되는 것을 확인할 수 있다.

* + **BC9**

|  |
| --- |
|  |
| **그림 13. testbench 코드 (3)** |
|  |
| **그림 14. 시뮬레이션 결과 (3)** |

위 그림은 4 bit BCD counter에 modulo 9 연산을 적용하여 0부터 9까지 카운트하는 BCD counter에 대한 시뮬레이션 결과이다. 위에서 기술한 BC6와 동일한 원리로 작동하며, 정상적으로 동작하는 것을 확인할 수 있다.

* **데모영상**
  + - [**PS2KEYBOARD**](https://youtu.be/9gd4xUP0iPk)
    - [**UART**](https://youtube.com/shorts/ODgpL-USB0U)