|  |  |
| --- | --- |
| **교과목 명** | 하드웨어 설계 |
| **설계 제목** | 다기능 전원 제어 장치 ( Multi-Function Power Supply Device ) |
| **설계자** | 김영찬 (2013122041), 이민우(2014122191) |
| **기간** | 2018년 11월 20일 화요일 – 2018 년 11월 27일 화요일 (1주차) |
| **장소** | 도서관 2층 세미나실 |
| **지도교수** | 이재환 |
| **논의** **내용** | 1. User Controller   1. 8 bit를 전송하기로 결정함. 기존에는 8 bit중 2 bit만 Preamble로 사용했지만, 회의 결과 Preamble 4bit와 데이터 4bit를 전송하기로 결정. 2. Preamble은 (0101)로 설정하였다. 3. Controller 제작에 필요한 소자들과 설계도를 Logic Works 5로 작성하였다.     2. GPIO 인터럽트 제어   1. C언어를 기반으로 코드를 작성 했을 때, pushButton을 눌렀을 때, GPIO에 연결된 LED를 제어함. 2. GPIO에 연결된 PushButton을 통해서 De1-SoC의 LEDR을 제어함 3. (1)번과 (3)번의 동작을 인터럽트로 동작하도록 함.     3. User Key Input Buffer 설정   1. 해당 프로젝트에는 De1-SoC의 Push Button과 Slide Switch 뿐 아니라 User Controller라는 외부 장치도 사용함. 해당 장치는 특수한 기능을 수행하는것이 아니라, De1-SoC에서 어떤 버튼을 눌렀는지 알려주는 장치임. 2. GPIO 인터럽트를 통해서 어떤 버튼이 눌렸는지를 알려주기 위해, De1-SoC내에서 사용할 수 있는 사용자 입력 버퍼를 만듦. 3. 인터럽트 발생시, 인터럽트 핸들러가 하는 일은 사용자 입력 버퍼에 입력한 키 값을 넣는것임. 4. De1-SoC 올리는 프로그램은, 주기적으로 버퍼를 확인하여, 버퍼에 값이 들어 있으면, 해당 값을 꺼냄. 5. 이것은 마치 키보드 자체에 기능을 넣는게 아니라, 키보드는 버퍼에 값을 넣고, 소프트웨어가 프로그래밍으로 ‘키보드의 ~키가 눌렸다면,’ 처럼 설정하는 것과 같은 이치임. |
| **진행 내용** | 1. User Controller   1. 사용자가 De1-SoC를 좀더 정밀하게 조작 할 수 있게 하는 컨트롤러를 제작함. 2. 컨트롤러의 구상도는 아래와 같음.      1. 컨트롤러는 총 16개의 버튼으로 구성되어 있으며, 각 버튼의 기능은 다음과 같음.  * 숫자 버튼 (10개) : 0 부터 9 까지의 숫자를 입력 할 수 있는 버튼 * OK 버튼 ( 1개 ) : 사용자의 입력을 마치거나 선택할 때 사용하는 확인 버튼 * BACK 버튼 ( 1개 ) : 사용자의 입력을 지우거나 선택을 해제할 때 사용하는 취소 버튼 * 방향 버튼 (4개) : 커서를 움직이거나, 올리거나 내릴 때 사용하는 UP, DOWN, LEFT, RIGHT 버튼  1. 컨트롤러는 16개의 버튼별로 각자 다른 8bit 코드를 Clock에 맞춰 전송함.  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 0번 키 | **0101** **0000** | 4번 키 | **0101 0100** | 8번 키 | **0101 1000** | LEFT 키 | **0101 1100** | | 1번 키 | **0101 0001** | 5번 키 | **0101 0101** | 9번 키 | **0101 1001** | RIGHT키 | **0101 1101** | | 2번 키 | **0101 0010** | 6번 키 | **0101 0110** | UP 키 | **0101 1010** | OK 키 | **0101 1110** | | 3번 키 | **0101 0011** | 7번 키 | **0101 0111** | DOWN키 | **0101 1011** | BACK 키 | **0101 1111** |  1. 컨트롤러의 전체적인 설계도는 아래와 같음.      1. 위 설계도의 전체적인 기능을 아래와 같음.  * 스위치 : Pull Up 저항을 사용하여 항상 5V를 인가하고, 버튼을 누르면 0V가 인가되도록 설계함. * Octal D F/F : 사용자의 입력을 잠시 저장 할 수 있도록 설계한 버퍼이다. Positive Egde Trigger를 사용함. * 74HC148 : 8 TO 3 Encoder이다. 2개를 사용하여 16 TO 4 Encoder로 사용함. 사용자 버튼이 총 16개이므로, 4 Bit 출력으로 Encoding이 가능함. * 74HC08 , 74HC04 : 8 TO 3 Encoder를 16 TO 4 Encoder로 합치는 과정에 사용 됨. * 74HC30 : 사용자가 버튼을 아무 버튼이나 눌렀을 때, SH/LD 가 활성화 되어야 하므로 버튼의 출력을 합치는 과정에 사용됨 * 74HC02 : 사용자가 버튼을 눌렀는지를 확인하는 비트를 합치는 용도로 사용됨 * 74HC165 : 컨트롤러에서 제일 중요한 8 bit Parallel Input Serial Output Shift Register임. 클락에 맞춰서 한 비트씩 옆으로 전송한다.  1. 컨트롤러와 De1-SoC는 아래와 같이 RJ45( 8 Pair ) 규격의 UTP Cable 을 사용한다.     2. GPIO 제어   1. De1-SoC의 KEY interrupt를 사용하여 GPIO에 연결된 LED를 제어함. 2. C기반의 main에서 KEY interrupt를 설정하고 NIOS ||에서 interrupt를 받을 수 있게 status를 1로 설정하고 while문 안에서 interrupt가 들어 올 때 까지 대기함. 3. JP1의 address 0xFF200060의 base는 쓰거나 읽을 수 있는 data register이고, 0xFF200064는 0이면 input, 1이면 output으로 설정하는 direction register인데, JP1의 핀을 이용해 연결된 LED를 제어하기 위해서 JP1을 제어 하는 것이므로 direction register를 모두 1로 설정해 출력으로 사용함. 4. JP1 40-pin expansion headers의 pinout은 다음과 같음.      |  |  | | --- | --- | | Clock\_in(pin 1) | D0 | | Clock\_in | D1 | | D2 | D3 | | D4 | D5 | | D6 | D7 | | VCC5 | GND | | D8 | D9 | | D10 | D11 | | D12 | D13 | |  | D14 | | D15 | | D16 | D17 | | D18 | D19 | | D20 | D21 | | VCC3.3 | GND | | D22 | D23 | | D24 | D25 | | D26 | D27 | | D28 | D29 | | D30 | D31(pin 40) |  1. VCC5를 통해 bread board의 100Ω resistor 와 RED light LED를 직렬로 연결하고 GND에 오도록 하면 낮은 resistor에 의해 RED light LED가 계속 켜져 있는 것을 확인함. 2. KEY0 또는 KEY1을 누르면 interrupt가 발생하고 KEY0를 누르면 GPIO의 D0가 HIGH로 설정되어 LED가 ON이 되게하고, KEY1을 누르면 D0가 LOW로 설정되어 LED가 OFF가 되게 함. |
| **진행 계획** | 1. User Key Input Buffer 설정   1. GPIO에 컨트롤러를 연결하여, GPIO인터럽트를 통해 입력 버퍼에 원하는 키를 넣는 ISR을 설계할 예정임. 2. 버퍼는 링 형태로 제작하며, 프로그램 어디에서든 참고 할 수 있도록 전역변수로 설정할 예정임. 3. 간단하게 Character형 ( 1 Byte ) 배열 ( Array )로 구현하도록 하며, 참고할 수 있도록 Out과 In이라는 변수를 선언하여 Pop과 Push를 할 수 있도록 할 예정임. 4. 프로그램은 주기적으로 버퍼를 관찰하여 버퍼에 값이 들어있다면 그 값을 Pop하여 버퍼에서 지우고, 해당 값에 맞는 동작을 실행하도록 설계할 예정임.   2. monitor에 TIMER와 상태 표시   1. 입력해준 TIMER에 따라 모니터에 TIMER가 작동 하는 것을 확인하고 색깔은 검정색으로 나타냄. 2. multi-tap의 각 port가 ON인지 OFF인지 나타내는데, ON이면 초록색, OFF이면 빨강색으로 나타냄. |
| **데모** **내용** | 1.KEY interrput에 의한 LED 전원 제어  1-1. KEY0를 누르면 LED ON, KEY1을 누르면 LED OFF 함.  1-2. bread board에 100Ω resistor 와 RED light LED를 직렬로 꽂고 JP1의 D0를 입력으로 줌.  1-3. 평가항목   1. NIOS ||의 pushbutton interrupt가 제대로 설정이 되었는가? 2. KEY interrupt 발생이 정상 작동 하는가? 3. KEY press에 의해 LED의 전원 제어가 되는가? |