Đại học Bách Khoa TP.HCM
Khoa Điện – Điện Tử
Bộ môn ĐKTĐ
---o0o---

ĐỀ THI HỌC KỲ 1. NH 2019-2020
Môn: **Hệ thống điều khiển nhúng – EE3067**
Ngày thi: 30/12/2019
Thời gian làm bài: 80 phút
(Sinh viên được phép sử dụng tài liệu)

Câu 1: (L.O.1) (2,5đ)

Thiết kế mạch giải mã địa chỉ và kết nối dữ liệu cho hệ thống (8-bit địa chỉ A7-A0; 8-bit dữ liệu D7-D0; RD, WR trên 2 chân riêng biệt) kết nối với các module bên dưới. Ghi rõ kết nối chân CS, WR, RD tới các module và địa chỉ bắt đầu, địa chỉ kết thúc của các module (không giải mã địa chỉ các kênh):

1/ Module 1: 16 kênh ADC 16-bit
2/ Module 2: 16 kênh Encoder 32-bit
3/ Module 3: 16 kênh PWM 10-bit
4/ Module 4: 4 kênh SPI 16-bit
5/ Module 5: 4 kênh Digital Input 8-bit
6/ Module 6: 2 kênh Digital Output 8-bit

Câu 2: (L.O.2) (2,5đ)

Viết chương trình Verilog thực hiện đo tốc độ động cơ bằng cách đo thời gian 4 chu kỳ của tín hiệu xung encoder:

- 3 đầu vào: 2 tín hiệu xung encoder: ENCA, ENCB; 1 tín hiệu xung clock 1us: clk.
- Đầu ra dữ liệu 16 bit: D[15:0], trong đó D[15] qui định chiều quay của động cơ, D[14:0] chứa thời gian (đơn vị us) của 4 chu kỳ xung ENCA.
- Dữ liệu ngõ ra được cập nhật sau mỗi 4 chu kỳ xung encoder ENCA.

Câu 3: (L.O.5) (2,5đ)

Trong yêu cầu thiết kế hệ thống dùng STM32F4 ta cần sử dụng Port {PD8, PD9} cho UART, {PE5, PE6} cho PWM, và 2 kênh DAC. Hãy cấu hình các thanh ghi hệ thống để sử dụng 3 chức năng này và UART có khả năng hoạt động DMA với các yêu cầu sau:

- Chỉ rõ chân nào sử dụng 2 kênh DAC
- Cấu hình thanh ghi cho phép xung clock để hoạt động các chức năng trên
- Cấu hình thanh ghi chân IO để lựa chọn chức năng phù hợp

Câu 4: (L.O.6) (2,5đ)

Trong cấu hình ở Câu 1, giả sử dùng STM32F4 để giải mã địa chỉ 8-bit dữ liệu và địa chỉ bắt đầu của CS_PWM là 0x6000_0000. Hàm nhận dữ liệu từ UART được cho trước như sau:

- Hàm nhận *N* byte từ bộ đệm *rxbuff* qua UART: void ***receive_data*** (char **rxbuff*, int *N*). Biết dữ liệu *rxbuff* dạng binary với tổng số byte nhận là 36 và format chuỗi dữ liệu nhận vào:

0xAA (1byte)	0x55 (1byte)	PWM1 (byteH)	PWM1 (byteL)	---	PWM16 (byteH)	PWM16 (byteL)	CRC (byteH)	CRC (byteL)

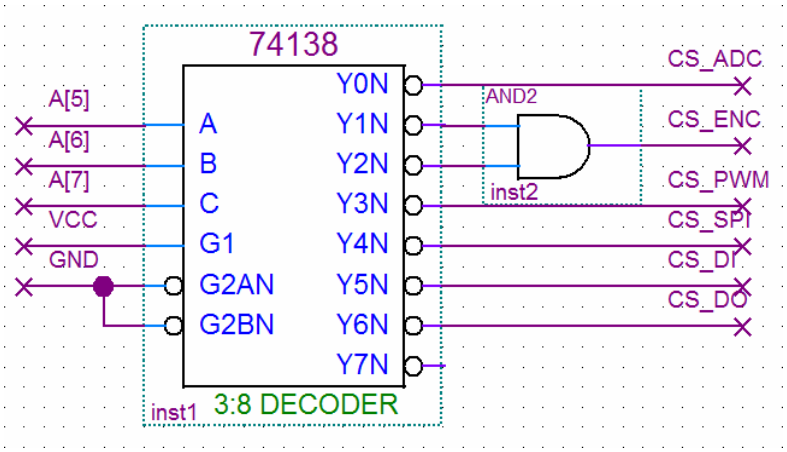
Viết chương trình đọc dữ liệu từ UART, kiểm tra chuỗi dữ liệu nếu header bằng {0xAA, 0x55} và đúng CRC thì chuyển đổi giá trị và xuất ra 16 kênh PWM, biết rằng CRC bằng tổng 34 byte dữ liệu trong frame (byte thứ 1 đến byte thứ 34).

CHỦ NHIỆM BỘ MÔN **GIÁO VIÊN RA ĐỀ**

ĐÁP ÁN

Câu 1: (2,5đ)

- Ghi tầm địa chỉ đúng cho 6 module và sơ đồ kết nối 74138 **(1.5đ)**



- Vẽ sơ đồ giải mã đúng 6 module **(1đ)**

Câu 2: (2,5đ)

- Khai báo đúng module, đầu vào, đầu ra **(0.5đ)**
- Đọc đúng thời gian (đơn vị us) cho 4 chu kỳ liên tiếp ENCA **(1đ)**
- Đọc đúng chiều quay và cập nhật ngõ ra sau 4 chu kỳ ENCA **(1đ)**

Câu 3: (2,5đ)

- Chỉ rõ chân kết nối: **(0.5đ)**
  PD8 – PD9: UART3 – AF7
  PE5 – PE6: PWM, TIM9 – AF3
  PA4 – PA5: DAC

- Cấu hình thanh ghi xung clock **(1đ)**

```
RCC_AHB1ENR |= (1<<0) | (1<<3) | (1<<4);        //clock cho PA, PD, PE
RCC_AHB1ENR |= (1<<21);                         //clock cho DMA1

RCC_APB1ENR  |= (1<<18)| (1<<29);               //clock cho UART3, DAC
RCC_APB2ENR  |= (1<<16);                        //clock cho PWM TIM9
```

- Cấu hình mode chức năng **(1.0đ)**

```
GPIOD_MODER |= (2<<16) | (2<<18);
GPIOE_MODER |= (2<<10) | (2<<12);
GPIOA_MODER |= (3<<8) | (3<<10);

GPIOD_AFRH   |= (7<<0) | (7<<4);
GPIOE_AFRL   |= (3<<20) | (3<<24);

DMA_S1CR     |= (4<<25);          //DMA1, Stream1, Channel4, UART3_RX
DMA_S3CR     |= (4<<25);          //DMA1, Stream3, Channel4, UART3_TX
```

Câu 4: (2,5đ)

```
char rxbuff[36];
uint8_t *p;
uint16_t crc, i;

main(){
  while(1){
    receive_data(rxbuff,36);
    crc = 0;
    for(i=0; i<34; i++)    crc += rxbuff[i];
    if ((rxbuff[0]==0xAA)&&(rxbuff[1]==0x55)&&
        (rxbuff[34]==(crc>>8))&&(rxbuff[35]==(crc&0xFF) )
    {
      p = 0x60000000;
      for(i=2; i<34; i++) *p++ = rxbuff[i];
    }
  }
}
```