Giảng viên ra đề: *(Chữ ký và Họ tên)*	*(Ngày ra đề)*	Người phê duyệt: *(Chữ ký, Chức vụ và Họ tên)*	*(Ngày duyệt đề)*

TRƯỜNG ĐH BÁCH KHOA – ĐHQG-HCM KHOA ĐIỆN ĐIỆN TỬ	**THI CUỐI KỲ**	**Học kỳ/năm học**	**1**	**2023-2024**
		Ngày thi	**23/12/2023**	
	Môn học	**Hệ thống điều khiển nhúng**		
	Mã môn học	**EE3067**		
	Thời lượng	**90 phút**	**Mã đề**	
Ghi chú: - Được 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 (16-bit địa chỉ A15-A0; 16-bit dữ liệu D15-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: 8 kênh Encoder 32-bit
3/ Module 3: 16 kênh PWM 10-bit
4/ Module 4: 2 kênh SPI 16-bit
5/ Module 5: 4 kênh Digital Input 8-bit
6/ Module 6: 8 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 liên tiếp 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 liên tiếp.
- Dữ liệu ngõ ra D[15:0] 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, {PE9, PE11} 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)

Giả sử dùng STM32F4 giải mã địa chỉ 16-bit dữ liệu để đọc dữ liệu 2 kênh encoder 32-bit như ở cấu hình Câu 1, biết địa chỉ bắt đầu của CS_ENC là 0x6000_0020.

Hàm xuất dữ liệu ra UART được cho trước như sau: void ***send_data*** (char *txbuff, int N). Trong đó *N* là số byte cần gửi từ bộ đệm *txbuff* ra UART

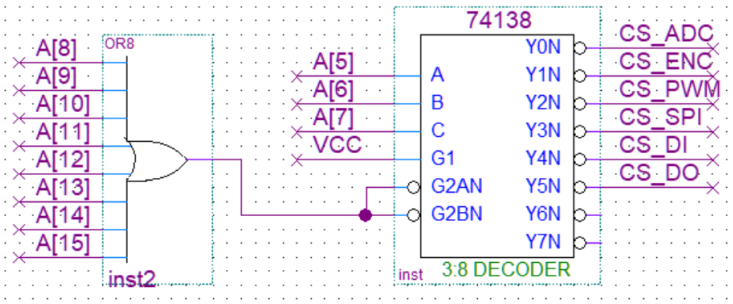
Viết chương trình đọc dữ liệu từ module Encoder, kênh 1 và 2, sau đó chuyển đổi giá trị int sang ASCII và gửi ra UART theo format chuỗi dữ liệu như sau biết rằng giá trị đọc encoder có tầm từ 0 - 200.000:

Encoder 1 (6byte)	0x20 (1byte)	Encoder 2 (6byte)	0x0D (1byte)	0x0A (1byte)

ĐÁP ÁN

Câu 1: (2,5đ)

- Vẽ đúng sơ đồ giải mã 74138 **(0.5đ)**



- Ghi đúng tầm địa chỉ cho 6 module **(1đ)**
- Vẽ đúng sơ đồ kết nối 6 module **(1đ)**

- Chú ý: Giải mã cho 16-bit data D[15:0], 16-bit địa chỉ A[15:0],
Nếu sinh viên làm giải mã 8-bit data D[7:0] = -1đ, 8-bit địa chỉ A[7:0] = -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: **(0.25đ)**
  - PD8 – PD9: UART3 – AF7
  - PE9, PE11: PWM, TIM1 – AF1
  - PA4 – PA5: DAC

- Cấu hình thanh ghi xung clock **(0.75đ)**

```
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<<0);                         //clock cho PWM TIM1
```

- Cấu hình mode chức năng **(1.5đ)**

```
GPIOD_MODER |= (2<<16) | (2<<18);
GPIOE_MODER |= (2<<18) | (2<<22);
GPIOA_MODER |= (3<<8) | (3<<10);

GPIOD_AFRH   |= (7<<0) | (7<<4);
GPIOE_AFRH   |= (1<<4) | (1<<12);

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

Câu 4: (2,5đ)

```
void IntToStr6(int32_t u, uint8_t *y){
  int32_t a, i;

  a = u;
  for (i=5; i>=0; i--){
    y[i] = a % 10 + 0x30;
    a = a / 10;
  }
}
main(){
  int32_t enc_value;
  unt16_t *p;
  uint8_t txbuff[15];

  p = 0x60000021;   //dia chi kenh 1 (MSB)
  enc_value = *p--;
  enc_value = enc_value<<16 + *p;
  IntToStr6(enc_value, txbuff);
  txbuff[6] = 0x20;

  p = 0x60000023;   //dia chi kenh 2 (MSB)
  enc_value = *p--;
  enc_value = enc_value<<16 + *p;
  IntToStr6(enc_value, txbuff+7);

  txbuff[13] = 0x0D;
  txbuff[14] = 0x0A;
  send_data(txbuff, 15);
}
```