ECE 355 | Fall 2024
Assignment 1

Question 1

```
#define PBIN (volatile unsigned char *)0xFFFFFFF3
#define PBOUT (volatile unsigned char *)0xFFFFFFF4
#define PBDIR (volatile unsigned char *)0xFFFFFFF5
#define PSTAT (volatile unsigned char *)0xFFFFFFF6
#define CNTM (volatile unsigned int *)0xFFFFFFD0
#define CTCON (volatile unsigned char *)0xFFFFFFD8
#define CTSTAT (volatile unsigned char *)0xFFFFFFD9
#define IVECT (volatile unsigned int *)(0x20)

interrupt void intserv();

volatile unsigned char digit = 0;       /* digit for display */

int main()
{
    unsigned char sample = 0;           /* Port B input sample */

    *PBDIR = 0b11110000;                /* Set Port B direction */
    *CTCON = 0b10;                      /* if the Timer is running, then stop it */
    *CTSTAT = 0b0;                      /* Clear "Reached 0" flag */
    *CNTM = 100000000;                  /* Initialize 1s timeout */
    *IVECT = (unsigned int *)&intserv;  /* Setup interrupt vector */
    asm("MoveControl PSR,#0b1000000");  /* CPU responds to IRQ */
    *CTCON = 0b1;                       /* Start Timer, disable interrupts for now */
    *PBOUT = 0b0;                       /* Display 0 */

    while (1)
    {
        while ((*PSTAT & 0b100) == 0);  /* Wait for PBIN update */
        sample = *PBIN & 0b11;          /* Sample PBIN, isolate bits [1:0] */
        if (sample == 0b1)
        {                               /* E = 0, D = 1 */
            *CTCON |= 0b10000;          /* Enable Timer interrupts */
        }
        else if (sample == 0b10)
        {                               /* E = 1, D = 0 */
            *CTCON &= 0b11101111;       /* Disable Timer interrupts */
        }
    }

    exit(0);
}

interrupt void intserv()
{
    *CTSTAT = 0b0;                      /* Clear "Reached 0" flag */
    digit = (digit + 1) % 10;           /* Increment digit */
    *PBOUT = digit << 4;                /* Update display */
}
```

Question 2

```
#define PCONT (volatile unsigned char *)0xFFFFFFF7
#define CNTM (volatile unsigned int *)0xFFFFFFD0
#define CTCON (volatile unsigned char *)0xFFFFFFD8
#define CTSTAT (volatile unsigned char *)0xFFFFFFD9
#define IVECT (volatile unsigned int *)(0x20)
interrupt void intserv();
int main()
{
    char digit = 0;                    /* Digit to be displayed */
    *PBDIR = 0b11110000;               /* Set Port B direction */
    *IVECT = (unsigned int *)&intserv; /* Set interrupt vector */
    asm("MoveControl PSR, #0x40");     /* CPU responds to IRQ */
    *PCONT = 0b01000000;               /* Enable PBIN interrupts */
    *CTCON = 0b0010;                   /* Stop Timer */
    *CSTAT = 0b0000;                   /* Clear "reached 0" flag */
    *CNTM = 100000000;                 /* Initialize Timer */
    *PBOUT = 0b0000;                   /* Display 0 */
    while (1)
    {
        while ((*CTSTAT & 0x1) == 0);  /* Wait until 0 is reached */
        *CSTAT = 0x0;                  /* Clear "reached 0" flag */
        digit = (digit + 1) % 10;      /* Increment digit */
        *PBOUT = digit << 4;           /* Update display */
    }
    exit(0);
}

interrupt void intserv()
{
    unsigned char sample;              /* Port B input sample */
    sample = *PBIN & 0b0011;           /* Sample PBIN, isolate bits [1:0] */
    if (sample == 0b0001)
        *CTCON = 0b0001;               /* Start Timer */
    else if (sample == 0b0010)
        *CTCON = 0b0010;               /* Stop Timer */
}
```

Question 3

Let x denote the I/O device activity percentage to be determined.

The maximum I/O data rate for DMA transfer is $\mathbf{R_{I/O}/d_{I/O\text{-}DMA} = 256\ transfers/s}$. The total DMA cost is: $\mathbf{(x*256)(N_{DMA\text{-}start} + N_{DMA\text{-}end}) = x*230.4K\ cycles/s}$, accounting for both the start and end CPU cycles per transfer.

For polling, the maximum I/O data rate is $\mathbf{R_{I/O}/d_{I/O} = 16{,}384\ transfers/s}$. The polling cost depends on the device's readiness: $\mathbf{(x*16{,}384)N_{poll\text{-}ready} + ((1-x)*16{,}384)N_{poll\text{-}not\text{-}ready} = x*4.915M + 1.638M\ cycles/s}$, with more frequent polling when the device is ready.

Since the DMA cost is 1,000 times cheaper than the polling cost, we have: $\mathbf{1{,}000*(x*230.4K) = x*4.915M + 1.638M}$, which simplifies to $\mathbf{x \approx 0.0072}$ (i.e., 0.72%).