

Ejercicio 1a

Existe un único camino entre E/S_1 y la CPU, pero hay cache, por lo que CPU podría estar cumpliendo su ciclo de ejecución con lo que tiene en la cache mientras el DMAC atiende la transferencia, ya sea de *memoria* a E/S_1 o E/S_2 o viceversa.

Ejercicio 1b

El CPU puede trabajar normalmente usando la *memoria*₁ y el dispositivo E/S_1 mientras el DMAC puede hacer transferencias entre los dispositivos E/S_2 y E/S_3 desde y hacia la *memoria*₂. Recordar que un DMAC transfiere datos desde la memoria principal a un dispositivo y desde un dispositivo a la memoria principal, no hace transferencias memoria-memoria o dispositivo-dispositivo.

Ejercicio 2

- Programa del CPU:

```

polling:  MOV [CTRL], 0xFF
          CMP [STATUS], 0xAC
          JNE polling

```

- Pseudocódigo del dispositivo:

```

while (true){
    while (CTRL != 0xFF) {
        /*skip*/
    }
    STATUS = 0xAC;
    /*<Acci'on>*/
    CTRL = 0x00;
    STATUS = 0x00;
}

```

- $CTRL$ y $STATUS$ quedan en 0x00.

Ejercicio 3

```

while(true)

    while(SD_CTRL != 0x00FF && SD_CTRL != 0x00AA){
        /*skip*/
    }

    SD_STATUS=0x0001           //Dispositivo Ocupado
    if(SD_CTRL == 0x00FF){    //lectura
        readDATA()           //leo
        while(!is_ready()){  //Espero que termine de leer
            /*skip*/
        }
    }
    else{                      //escritura
        writeData();         //escribo
        while(!is_ready()){  //Espero que termine de escribir
            /*skip*/
        }
    }
}
SD_CTRL=0x0000               //Sino vuelvo a entrar en algun if
SD_STATUS=0x0000            //Aviso que termine
}

```

Ejercicio 4

```

While (true){
  While (DMA_CTRL[15] != 1) {                               //Miro el bit mas significativo
    /*skip*/
  }
  If (DMA_CTRL[0] == 1){                                     // Dispositivo a Memoria
    while(SIZE>0){

      writeIO (SD_IO_ADRESS, DMA_IO_ADRESS) //Escribo en el registro SD_IO_ADRESS del disp
      WriteIO (SD_CTRL, 0x00FF)             //escribo en el registro SD_CTRL del dispositivo

      While (readIO(SD_STATUS) != 0x0000){ //Espero que el disp termine
        /*skip*/
      }

      DATA=readIO(SD_DATA)                    //Traigo el valor de SD_DATA
      writeMEM(DMA_MEM_ADDRESS,DATA)          //Escribo el valor en memoria principal

      DMA_MEM_ADDRESS++
      DMA_IO_ADDRESS++
      DMA_SIZE--
    }
  }Else{                                                    // Memoria a Dispositivo
    while(SIZE>0){

      DATA = readMem(DMA_MEM_ADDRESS)        //leo de memoria el dato
      writeIO (SD_DATA, DATA)                //Escribo en el registro SD_DATA del disp
      writeIO (SD_CTRL, 0x00AA)               //escribo en el registro SD_CTRL del dispositivo

      While (readIO(SD_STATUS) != 0x0000){ //Espero que el disp termine
        /*skip*/
      }

      DMA_MEM_ADDRESS++
      DMA_IO_ADDRESS++
      DMA_SIZE--
    }
  }
  DMA_CTRL[15] = 0                                         //Para que no vuelva a entrar
  requestInt()                                             //Aviso al cpu que termine
}

```

Notar que se usó una variable llamada *DATA*, esto se puede interpretar como un buffer interno del DMAC donde se almacena un dato temporal.

Ejercicio 5

```

Inicio:  MOV [0xFFF1], 0x0045 ; [DEVICE_ADDRESS]
         MOV [0xFFF2], 0xA142 ; [MEM_ADDRESS]
         ;
         MOV [0xFFF3], 0x00F5 ; [SIZE]
         MOV [0xFFF4], 0x8000 ; [STATUS]

```

Notar que es importante que el registro *STATUS* sea el último en escribirse ya que con el se le indica al DMAC que comience la transferencia.