Arquitectura de Computadoras TIEMPO DB 0 TERMINAR DB 0 ORG 3000h RUT\_F10: PUSH AX MOV TERMINAR, 1 MOV AL, EOI OUT EOI, AL POP AX IRET ORG 3200h RUT\_CLK: PUSH AX MOV TIEMPO, 1 MOV AL, 0 OUT CONT, AL MOV AL, EOI OUT EDI, AL POP AX IRET ORG 3400h ESPERAR: CLI MOV AL, OFCh OUT IMR, AL MOV AL, 0 OUT CONT, AL MOV TIEMPO, 0 STI CMP TERMINAR, 1 LOOP: JZ SALIR CMP TIEMPO, 1 JNZ LOOP SALIR: CLI MOV AL, OFEH OUT IMR, AL STI RET ORG 2000h CLI MOV AL, OFCh OUT IMR, AL MOV AL, N\_F10 OUT INTO, AL MOV AL, N\_CLK OUT INT1, AL MOV AL, 0 OUT CONT, AL MOV AL, 5 OUT COMP, AL MOV AL, 0 OUT CB, AL STI MOV BX, OFFSET CADENA 25/26

mensaje

mensaje

mensaje

mensaje

oet 96

ayie dw mens

Arquitectura de Computadoras

2026

## Parte 3: Ejercicios de repaso para el parcial

Imprimir mensaje leido, longitud arbitraria \*\* Escribir un programa que solicite ingresar caracteres
por teclado y los almacene en la memoria. Al presionar la tecla F10, se envían a la impresora a través
de la P10. No es necesario mostrar los caracteres en la pantalla.

```
PA EQU 30h
PB EQU 31h
       CA EQU 32h
       CB EQU 33h
       IMR EQU 21h
      EOI EQU 20h
INTO EQU 24h
      N_F10 EQU 10
      ORG 40
      DW RUT_F10
     ORG 1000h
     FIN_LEER DB 0
     LARGO DB 0
     CADENA DB ?
    ORG 3200h
    RUT_F10: PUSH AX
              MOV AL, OFFH
OUT IMR, AL
              MOV FIN_LEER, 1
             MOV AL, EOI
OUT EOI, AL
             POP AX
             IRET
   ORG 3000h
   LEE_CADENA: CLI
               MOV AL, ØFEH
OUT IMR, AL
               MOV AL, N_F10
               OUT INTO, AL
               STI
               MOV BX, OFFSET CADENA
               MOV LARGO, 0
       SIGUE: INT 6
              INC BX
              INC LARGO
              CMP FIN_LEER, Ø
              JZ SIGUE
              RET
ORG 2000h
      CALL LEE_CADENA
     MOV AL, 0FDh ; strobe salida (0), busy entrada (1)
     OUT CA, AL
     MOV AL, 0 ; puerto de datos todo salida
```

		Sept of The Order of	
			La CPU está ejecutando el
	granding visite about	servicio. El bit que representa la linea de representa la linea de representa la linea de interrupción que se está atendiendo estará en 1 atendiendo estará en 1 interrupción (bit 0 se asocia a la entrada INTO bit 7 se asocia a la entrada INT7), el resto en 0. Si no hay interrupciones atendiéndose todos los bits valen 0.	manejador de interrupción INT2. Esa interrupción está conectada al HandShake.
24h	INTO	ID de Linea INTO. Almacena el 10 de la interrupción asociada al dispositivo F10 para buscar en el vector de interrupciones la dirección de comienzo de la subrutina que lo atiende.	Es el número de vector que se le enviará a la CPU para manejar la interrupción. Si multiplico por 4 (15×4=60) nos da la dirección de memoria donde debe estar la dirección de rutina que maneja la interrupción correspondiente a F10.
25h	INTI	ID de Linea INT1. Almacena el ID de la interrupción asociada al dispositivo Timer para buscar en el vector de interrupciones la dirección de comienzo de la subrutina que lo atiende.	Es el número de vector que se le enviará a la CPU para manejar la interrupción. Si multiplico por 4 (10×4=40) nos da la dirección de memoria donde debe estar la dirección de rutina que maneja la interrupción correspondiente a Timer.
26h	INT2	ID de Linea INT2. Almacena el ID de la interrupción asociada al dispositivo Handshake para buscar en el vector de interrupciones la dirección de comienzo de la subrutina que lo atiende.	Es el número de vector que se le enviará a la CPU para manejar la interrupción. Si multiplico por 4 (25×4=100) no da la dirección de memoria donde debe estar la dirección de rutina que maneja la interrupción correspondiente Handshake.

2) Configuración del PIC Se desean configurar algunas interrupciones mediante el PIC. Complete la tabla indicando en cada caso los registros o direcciones que faltan para que funcionen las interrupciones, o el dispositivo que se está usando.

Dispositivos	IMR	INTO	INT1	INT2	Dirección Vector
F10	1111 1110	5	No importa	No importa	20

```
MOV AL, OFFSET FIN_TEXTO - OFFSET TEXTO
           INT 7
          MOV AL, EOI
          OUT EOI, AL
          POP BX
          POP AX
          IRET
      ORG 2000H
      CLI
      MOV AL, OFEH
      OUT IMR, AL
      MOV AL, N_F10
      OUT INTO, AL
      STI
LAZO: JMP LAZO
END
```

B) Idem A pero ahora el programa termina luego de mostrar 5 veces el mensaje (es decir, luego de que presione 5 veces la tecla F10).



Arquitectura de Computadoras

Pieta: cuando se ejecuta la 5ta interrupción, deben deshabilitarse las mismas para evitar reaccionar a próximas pulsaciones de la tecla Pio. Utilizar una variable global para registrar cuantas veces faita mostrar el mensaje antes de terminar.

```
EOI EQU 20H
IMR EQU 21H
INTO EQU 24h
    ORG 1000H
TEXTO DB "Vamos las interrupciones!"
   FIN_TEXTO DB 7
CANT DB 0
   ORG 2000H
          CLI
MOV AL, 0FEH
OUT IMR, AL
MOV AL, N_F10
OUT INTO, AL
           STI
  LAZO: CMP CANT, 5
           JNZ LAZO
           INT Ø
    ORG 3000H
  RUT_F10: PUSH AX
              PUSH BX
              MOV BX, OFFSET TEXTO
              MOV AL, OFFSET FIN_TEXTO - OFFSET TEXTO
              INT 7
              INC CANT
CMP CANT,5
JNZ SIGUE
MOV AL, ØFFH
OUT IMR, AL
SIGUE: MOV AL, EOI
             OUT EOI, AL
             POP BX
             POP AX
             IRET
END
```

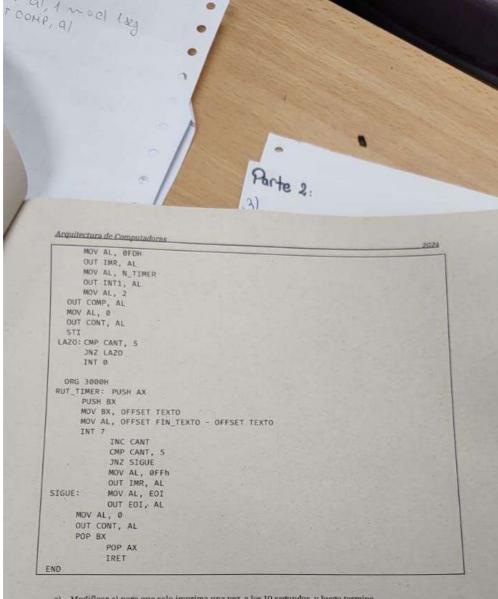
5) Timer, tres usos: periódico infinito, periódico finito, y única vez
 a) Escribir un programa que muestra el mensaje "Vamos las interrupciones" una vez cada 2 segundos y no termine nunca.

EOI EQU 20H IMR EQU 21H INT1 EQU 25h CONT EQU 10h COMP EQU 11h N\_TIMER EQU 20

```
ORG 80
IP_FIO DW RUT_TIMER
  ORG 1000H
TEXTO DB "Vamos las interrupciones!"
FIN_TEXTO DB ?
     ORG 2000H
       MOV AL, OF DH
       OUT IMR, AL
MOV AL, N_TIMER
OUT INT1, AL
MOV AL, 2
OUT COMP, AL
       MOV AL, 0
       OUT CONT, AL
       STI
 LAZO: JMP LAZO
  ORG 3000H
 RUT_TIMER: PUSH AX
       PUSH BX
       MOV BX, OFFSET TEXTO
MOV AL, OFFSET FIN_TEXTO - OFFSET TEXTO
       INT 7
       MOV AL, EOI
         OUT EOI, AL
       MOV AL, 0
       OUT CONT, AL
       POP BX
             POP AX
              IRET
END
```

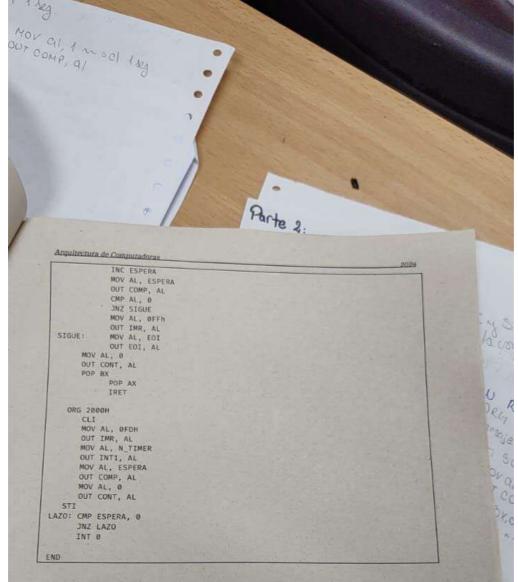
## b) Modificar a) para que termine a los 10 segundos.

```
EOI EQU 20H
IMR EQU 21H
INT1 EQU 25h
CONT EQU 10h
COMP EQU 11h
N_TIMER EQU 20
ORG 80
IP_F10 DW RUT_TIMER
 ORG 1000H
TEXTO DB "Vamos las interrupciones!"
FIN_TEXTO DB ?
CANT DB Ø
 ORG 2000H
    CLI
```



c) Modificar a) para que solo imprima una vez, a los 10 segundos, y luego termine.

```
EOI EQU 20H
 IMR EQU 21H
 INT1 EQU 25h
 CONT EQU 10h
 COMP EQU 11h
 N_TIMER EQU 20
 ORG 80
IP_F10 DW RUT_TIMER
TEXTO DB "Vamos las interrupciones!"
FIN_TEXTO DB ?
TERMINA DB 0
 ORG 3000H
RUT_TIMER: PUSH AX
    PUSH BX
```

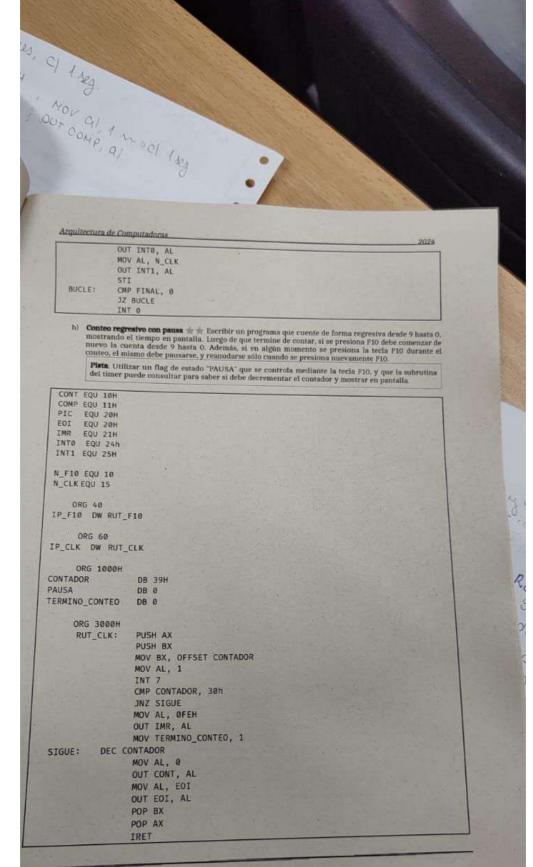


6) F10 con Timer
a) Conteo regresivo Escribir un programa que implemente un conteo regresivo a partir de un valor (de 1 a 9) ingresado desde el teclado. El conteo debe comenzar al presionarse la tecla F10. El tiempo transcurrido debe mostrarse en pantalla, actualizándose el valor cala segundo. Por ejemplo, si se ingresa el valor 7, cuando se aprieta F10 debe mostrarse en pantalla "7 6 5 4 3 2 1 0" en los 7 segundos siguientes

Pista: Una vez leido el digito se puede activar solamente la interrupción F10, y cuando se invoque la subrutina F10 activar el Timer para que a partir de ese momento se muestre el conteo regresivo.

```
CONT EQU 10H
 COMP EQU 11H
 PIC EQU 20H
EOI EQU 20H
IMR EQU 21H
 INTO EQU 24h
INT1 EQU 25H
N_F10 EQU 10
N_CLK EQU 15
     ORG 40
IP_F10 DW RUT_F10
```

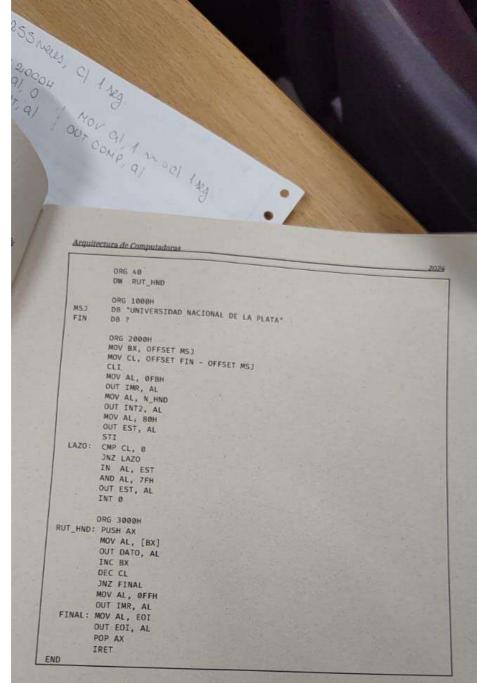
2550000, 0/ 1823 mov al, I would Arquitectura de Computadoras ORG 60 IP\_CLK DW RUT\_CLK ORG 1000H CONTADOR DB 39H FINAL DB 0 ORG 3000H PUSH AX RUT\_CLK: PUSH BX MOV BX, OFFSET CONTADOR MOV AL, 1 INT 7 CMP CONTADOR, 30h JNZ SIGUE MOV AL, ØFFH OUT IMR, AL MOV FINAL, 1 SIGUE: DEC CONTADOR MOV AL, 0 OUT CONT, AL MOV AL, EOI OUT EOI, AL POP BX POP AX IRET\_ ORG 3200H RUT\_F10: PUSH AX MOV AL, ØFDH OUT IMR, AL MOV AL, Ø OUT CONT, AL MOV AL, 1 OUT COMP, AL MOV AL, EOI OUT EOI, AL POP AX IRET ORG 3400H MOV BX, OFFSET CONTADOR LEE\_NUMERO: INT 6 REPITE: MOV AL, CONTADOR CMP AL, 30H JS REPITE CMP AL, 39H JNS REPITE RET ORG 2000H CALL LEE\_NUMERO CLI MOV AL, ØFEH OUT IMR, AL MOV AL, N\_F10 10/26



```
d. Imprimin HSS 255 West, Of Long
    Arquitectura de Computadoras
        ORG 3200H
         RUT_F10: PUSH AX
CMP TERMINO_CONTEO, 1
                  JNZ NO_REPETIR
                  MOV TERMINO_CONTEO, Ø
                  MOV CONTADOR, 39H
                 MOV AL, Ø
OUT CONT, AL
                  MOV AL, OFCh
                  JMP CONT_F10
     NO_REPETIR: CMP PAUSA, 0
                 JZ PAUSAR
                 MOV PAUSA, 0
                 MOV AL, ØFCH
                 JMP CONT_F10
         PAUSAR: MOV PAUSA, 1
                MOV AL, 0
                 OUT CONT, AL
                MOV AL, ØFEh
       CONT_F10:OUT IMR, AL
                MOV AL, EOI
OUT EOI, AL
                POP AX
                IRET
  ORG 2000H
   CLI
   MOV AL, ØFCH
   OUT IMR, AL
MOV AL, N_F10
   OUT INTO, AL
   MOV AL, N_CLK
   OUT INT1, AL
   MOV AL, 0
   OUT CONT, AL
     MOV AL, 1
   OUT COMP, AL
   STI
BUCLE: JMP BUCLE
```

7) Impresión con HANDSHAKE mediante interrupciones \*\*\*
a) Escribir un programa que imprime "UNIVERSIDAD NACIONAL DE LA PLATA" en la impreso través del HAND-SHAKE. El envío de los caracteres se realiza por interrupciones emitidas desd HAND-SHAKE cada vez que detecta que la impresora se desocupa.

IMR EQU 21H EOI EQU 20H INT2 EQU 26H HAND EQU 40H EST EQU 41H DATO EQU 40H N HND EQU 10



b) Probar el programa utilizando:
 1) Una velocidad de reloj del 100%, y una velocidad de impresión del 25%
 2) Una velocidad de reloj del 25%, y una velocidad de impresión del 100% ¿En qué caso seria mejor utilizar consulta de estado en lugar de interrupciones?

En el caso de una CPU lenta y una impresora rápida (caso 2) la consulta de estado resulta más conveniente. La complejidad que añaden las interrupciones no se justifica si la espera es poca.

c) Modificar a) de modo que el string a imprimir se lea desde teclado. El string a leer tiene una longitud fija de 10 caracteres, y se lee de forma completa antes de comenzar la impresión. Tener en cuenta entonces que las interrupciones deben esperar a que se ingrese todo el string.

DATO EQU 40h EST EQU 41h