

los registros del TIMER:

a) Cómo funciona el TIMER y cuándo emite una interrupción a la CPU:

El **TIMER** es un componente del hardware que genera una interrupción periódica a la CPU a intervalos de tiempo específicos. En el contexto de este programa, se usa para generar una interrupción cada vez que transcurre un segundo, lo que permite contar los segundos transcurridos desde que se inicia la ejecución del programa.

- **Emisión de Interrupción:** El **TIMER** emite una interrupción cuando se llega a un valor de contador específico, generalmente a intervalos regulares definidos por un valor de registro de comparación (en este caso **TIMER+1**), lo que genera una señal de interrupción a la CPU.
- **Ciclo del TIMER:** Una vez que el registro de contador llega al valor de comparación, el **TIMER** genera una interrupción y reinicia el contador. Esto se configura utilizando el registro **TIMER+1** (el cual define el valor de comparación) y el registro **TIMER** (que almacena el valor actual del contador).
- **Interrupción:** Cuando el contador del **TIMER** llega a su valor de comparación, la interrupción es emitida, y el procesador realiza una transferencia al vector de interrupción correspondiente (en este caso **IP_CLK**), lo que ejecuta la rutina de interrupción **RUT_CLK** para actualizar los segundos.

b) La función que cumplen los registros, la dirección de cada uno y cómo se programan:

1. **TIMER** (Dirección: 10H):

- Este registro es usado para almacenar y manejar el contador del timer. Al llegar a su valor de comparación, genera una interrupción.

2. **PIC** (Dirección: 20H):

- Es el **Controlador de Interrupciones Programable** que gestiona la distribución de las interrupciones a la CPU. En este programa, el PIC controla la habilitación de la interrupción generada por el **TIMER**.
- **IMR (Dirección: 21H):** Es el registro de máscara de interrupciones. Permite habilitar o deshabilitar interrupciones específicas.
- **INT1 (Dirección: 25H):** Este registro se usa para especificar la interrupción que se activará para el **TIMER**. En este caso, el número **N_CLK** (10H) es escrito en **INT1** para habilitar la interrupción del **TIMER**.

3. **EOI (End of Interrupt)** (Dirección: 20H):

- Es el registro que la CPU utiliza para enviar una señal de fin de interrupción. Se escribe en este registro cuando se ha manejado la interrupción, permitiendo que el PIC sepa que ha terminado de procesar la interrupción, y pueda proceder a la siguiente.

4. **TIMER+1** (Dirección: 11H):

- Este es el **registro de comparación** que establece el valor al cual el contador del **TIMER** debe llegar para generar una interrupción. Cuando el contador del **TIMER** llega a este valor, se genera una señal de interrupción.

5. **TIMER** (Dirección: 10H, Registro **CONT**):

- Este es el **registro de contador** que va aumentando continuamente y que se compara con el valor en **TIMER+1**. Al llegar a su valor de comparación, emite una interrupción.

Programación de los registros:

- **Inicialización del PIC:**

```
MOV AL, 0FDH
OUT PIC+1, AL ; Deshabilitamos las interrupciones del PIC para habilitar sólo las necesarias.
MOV AL, N_CLK
OUT PIC+5, AL ; Especificamos que la interrupción que debe ser habilitada es la número N_CLK.
```

- Configuración del TIMER:

```
MOV AL, 1
OUT TIMER+1, AL ; Configuramos el valor de comparación del TIMER para que genere una interrupción cada vez que el contador llegue a este valor.
MOV AL, 0
OUT TIMER, AL ; Inicializamos el valor del contador del TIMER a 0.
```

- Habilitación de interrupciones:

```
STI ; Habilitamos las interrupciones globalmente para que se puedan generar interrupciones por el TIMER.
```

- Rutina de Interrupción `RUT_CLK`: Esta rutina se ejecuta cuando se emite la interrupción del TIMER. Aquí es donde se actualiza el valor de los segundos y se muestra el contador en pantalla.

```
RUT_CLK: PUSH AX
INC SEG+1 ; Incrementamos los segundos (valor de SEG+1).
CMP SEG+1, 3AH ; Comprobamos si el valor de SEG+1 ha alcanzado '3A' (es decir, 60 en decimal).
JNZ RESET ; Si no ha alcanzado 60, saltamos al reinicio del contador.
MOV SEG+1, 30H ; Reiniciamos el contador de segundos.
INC SEG
CMP SEG, 36H ; Si el valor de SEG alcanza '36' (es decir, 6 en hexadecimal), reiniciamos el contador de minutos.
JNZ RESET ; Si no ha llegado a 6 (es decir, 60 segundos), reiniciamos el contador de segundos.
MOV SEG, 30H ; Reiniciamos el contador de segundos a 0.
RESET:
INT 7 ; Finalizamos la interrupción y pasamos al siguiente ciclo.
MOV AL, 0 ; Limpiamos la señal de interrupción.
OUT TIMER, AL ; Reiniciamos el contador del TIMER.
MOV AL, EOI ; Indicamos el fin de la interrupción.
OUT PIC, AL ; Finalizamos la interrupción en el PIC.
POP AX ; Restauramos el valor de AX.
IRET ; Retornamos del manejador de interrupciones.
```

Conclusión:

El **TIMER** se utiliza en este programa para emitir una interrupción cada segundo, y la rutina asociada a esta interrupción actualiza y muestra el número de segundos transcurridos en pantalla, desde el inicio de la ejecución. La configuración del **TIMER** y el **PIC** son esenciales para manejar las interrupciones correctamente, asegurando que el reloj funcione de manera precisa y continua.