TIMERS

Saturday, October 2, 2021 12:44

- Esta placa tiene 4 timers: TIMER0-3
- Son de 32 bits
- La señal que le llega a los timers (PCLK) pasa primero por un prescaler para acondicionar la señal
- PCLK no es necesariamente la señal que le llega al CPU:
  - PCLK es CCLK (la señal de clock que le llega al CPU) pasada por un divisor de frecuencia (puede ser CCLK, CCLK/2, etcétera)
- El prescaler es un contador ascendente, y tiene un registro asociado (PRESCALER REGISTER - PR) de 32 bits
  - Cuando el valor de cuenta del prescaler coincida con el valor en el registro PR, el prescaler se resetea (vuelve a 0) y esto genera un pulso que le llega al timer para que incremente en 1 su cuenta
  - El valor de división aplicado al timer es PR + 1
    - Si PR = 0, estamos usando CCLK
    - Si PR = 1, estamos usando CCLK/2
    - Si PR = 2, estamos usando CCLK/3
    - …

MODO MATCH

- En esta placa tenemos 4 registros con los que podemos comparar el valor del timer (MATCH REGISTER 0-3 = MR0-3 = MAT0-3) para, por ejemplo, cuando se cuente cierta cantidad de tiempo, generar un evento (una interrupción, frenar el timer, cambiar flags, etcétera)
- Cada timer tiene sus cuatro MR asociados
- El tiempo en el cual el timer se incrementará será: $T = \frac{(PR+1)(MR+1)}{PCLK}$
- El tiempo en el cual se van a producir interrupciones en este modo será: $T = \frac{L(PR+1)(MR+1)}{PCLK}$
  - Siendo L el valor cargado en match
- Registros asociados:
  - PCONP:
    - Para encender los timers
    - TIMER0-1 están habilitados por reset, pero TIMER2-3 no
    - <1> (PCTIM0):
      - 0 : TIMER0 off
      - 1 : TIMER0 on
    - <2> (PCTIM1):
      - 0 : TIMER1 off
      - 1 : TIMER1 on
    - <22> (PCTIM2):
      - 0 : TIMER2 off
      - 1 : TIMER2 on
    - <23> (PCTIM3):
      - 0 : TIMER3 off
      - 1 : TIMER3 on
  - PCLKSEL:
    - <3:2> (PCLK_TIMER0): Indica la división del clock de TIMER0
    - <5:4> (PCLK_TIMER1): Indica la división del clock de TIMER1
    - <13:12> (PCLK_TIMER2): Indica la división del clock de TIMER2
    - <15:14> (PCLK_TIMER3): Indica la división del clock de TIMER3
      - 00 : PCLK = CCLK/4
      - 01 : PCLK = CCLK
      - 10 : PCLK = CCLK/2
      - 11 : PCLK = CCLK/8
  - TXIR:
    - Controla las interrupciones del timer X
    - Cualquier MR me va a llevar a un mismo vector de interrupción, y allí dentro tengo que preguntar cuál MR fue
    - Se deben limpiar las flags antes de salir de la rutina de interrupción
  - TXTC:
    - Registro TIMERX propiamente dicho
  - TXPR:
    - Prescaler del timer X
  - TXPC:
    - Cuenta del prescaler del timer X
  - TXTCR:
    - Registro de control del timer X
    - Para habilitar o deshabilitar el timer X
      - <0> (ENABLE):
        - 0 : El timer X está deshabilitado
        - 1 : El timer X está habilitado
      - <1> (RESET):
        - 0 : El timer X y su prescaler no se resetean
        - 1 : El timer X y su prescaler se resetean (ambos registros se mantienen en 0 hasta que este bit pase a 0)
          - El timer no comenzará a contar hasta que este bit pase a 0
  - TXMRY:
    - MRY del timer X
  - TXEMR:
    - Registro de match externo del timer X
    - Para definir qué va a suceder con el pin de salida asociado para el match
      - 00 : No pasa nada
      - 01 : Pone el/los pin/es a 0
      - 10 : Pone el/los pin/es a 1
      - 11 : Togglea el/los pin/es
  - TXMCR:
    - Con este registro podemos configurar si cada vez que se genere un match, se genera una interrupción y si vamos a resetear el timer X; por ejemplo:
      - <0> (MR0I):
        - 0 : Cuando se produce un match con MR0, se produce una interrupción
        - 1 : Cuando se produce un match con MR0, no se produce una interrupción
      - <1> (MR0R):
        - 0 : Cuando se produce un match, el timer X se resetea
        - 1 : Cuando se produce un match, el timer X no se resetea
      - Ver demás bits de este registro en el manual de usuario
  - TXCTCR:
    - Seleccionamos si queremos que el registro TIMERX se comporte como contador o timer
- TIMER0-1 y TIMER3 tienen asociados un solo pin de salida para los matches (pines de match externo), pero TIMER2 tiene 4 pines asociados, donde el comportamiento especificado se replica en todos los pines
- $MatchTime[s] = PrescaleValue[s] * (MatchValue + 1)$

MODO CAPTURE

- Otra forma de uso es capturar el valor de cuenta del timer cuando ocurre algún evento externo en alguno de los pines asociados al timer (evento por flanco de subida o bajada)
  - Cuando ocurre el evento, se almacena una copia del valor de cuenta del timer en ciertos registros especiales
  - Estos registros especiales son CAPTURE REGISTER 0-1 = CR0-1 = CAP0-1
- El tiempo en el cual se van a producir interrupciones en este modo será: $T = \frac{C(PR+1)(CR+1)}{PCLK}$
  - Siendo C el valor capturado
- Registros de capture:

- CCR:
  - Configuramos el evento que va a disparar la captura (flanco de bajada o de subida)
- CAP0I:
  - Habilitamos que una interrupción se dispare cuando ocurra ese evento
- TCR:
  - Habilitamos el timer y lo ponemos en reset