ADC

Thursday, September 30, 2021 19:42

- 12 bits
  - Tenemos 4096 posiciones (2^{12})
  - La resolución será de $\frac{3.3}{4096}[V]$
- Aproximaciones sucesivas
- Soporta DMA (puedo grabar datos en memoria sin usar el core del micro)
- Se puede multiplexar hasta 8 pines (puedo leer y convertir hasta 8 entradas analógicas)
- Trabaja con 3.3[V] (si se trabaja con más tensión, el fabricante no garantiza correcta funcionalidad)
- Soporta hasta 5[V] si se usa como GPIO
- Convierte hasta a 200[KHz] (frecuencia máxima de muestreo - 200000 muestras por segundo)
  - La frecuencia máxima de trabajo posible para el ADC es de 13[MHz]
  - Con estos datos, si hacemos $\frac{13[MHz]}{200[KHz]}$ obtenemos que se necesitarían 65 pasos de conversión
  - En modo de trabajo de no - ráfaga se necesitan 65 clocks para convertir
  - En modo de trabajo de ráfaga se necesitan 64 clock para convertir
- Puede trabajar con conversión burst (conversión permanente)
- Para usar cualquier periférico con el ADC hay que:
  - Encender el periférico
  - Configurar el clock del periférico (ver PCLKSEL)
  - Configurar las funciones del pin (PINSEL)
- Cuando utilizamos un pin para el ADC, no sólo tenemos que modificar su función en PINSEL sino que también ponerlo en modo *neither pull up nor pull down* porque si tenemos alguna configurada va a ocasionar errores en la medición del valor
- Registros asociados:
  - PCONP:
    - <12> (PCADC):
      - 0 = ADC no alimentado
      - 1 = ADC alimentado
  - PCLKSEL:
    - <25:24> (PCLK_ADC):
      - Indican a qué frecuencia trabaja el ADC
      - El clock que se está dividiendo puede ser el clock interno de la placa o un clock externo que se utilice
      - 00 = CCLK/4
      - 01 = CCLK
      - 10 = CCLK/2
      - 11 = CCLK/8
  - ADCR:
    - <7:0> (SEL):
      - 0 = ADDRX no está convirtiendo
      - 1 = ADDRX está convirtiendo
        - Si se trabaja en modo ráfaga, podemos tener varios canales habilitados para convertir
        - Si no se trabaja en modo ráfaga, sólo puede haber un canal habilitado para convertir
        - Los pines configurados como inputs para el ADC no deben estar configurados como pull - up ni pull - down, ya que dicha configuración puede generar errores en las lecturas
        - Si el ADC muestrea a N[KHz] y estamos usando un solo canal, ese canal convertirá N[KHz]
        - Si el ADC muestrea a N[KHz] y estamos usando M canales, cada canal convertirá a $\frac{N}{M}[KHz]$
    - <15:8> (CLKDIV):
      - Para dividir el clock que le llega al ADC
    - <16> (BURST):
      - 0 = Los bits de start activan la conversión del ADC (BURST mode off)
      - 1 = El ADC está continuamente convirtiendo (BURST mode on)
    - <20:17>:
      - Reservados
    - <21> (PDN):
      - 0 = ADC off
      - 1 = ADC on
    - <23:22>:
      - Reservados
    - <26:24> (START):
      - Discrimina el modo de trigger del ADC para comenzar a convertir
      - Si estoy en modo BURST, no tiene sentido configurar START porque la conversión va a ser permanente e ininterrumpida
      - 000 = El ADC no trabaja (si estoy en BURST, trabaja igualmente aunque ponga 000 acá)
      - 001 = Comienza a convertir
      - 010 = Una señal que entre por P2.10 (configurado como EINT0) será quien indique el comienzo de la conversión
      - 011 = Una señal que entre por P2.27 (configurado como CAP0.1) será quien indique el comienzo de la conversión
      - 100 = Un match con MAT0.1 será quien indique el comienzo de la conversión
      - 101 = Un match con MAT0.3 será quien indique el comienzo de la conversión
      - 110 = Un match con MAT1.0 será quien indique el comienzo de la conversión
      - 111 = Un match con MAT1.1 será quien indique el comienzo de la conversión
    - <27> (EDGES):
      - Elijo flaco trigger para conversión en caso de usar una configuración de START entre 010 y 111
      - 0 = Flanco ascendente
        1 = Flanco descendente
    - <31:28>:
      - Reservados
  - ADGDR:
    - Contiene el resultado de la conversión más reciente que tiene completada, e incluye copias de los indicadores de estado que acompañan a esa conversión
    - <3:0>:
      - Reservados
    - <15:4> (RESULT):
      - Resultado de la conversión completada más reciente
    - <23:16>:
      - Reservados
    - <26:24> (CHN):

            - 000 = La conversión completada más reciente fue hecha en el canal 0
            - 001 = La conversión completada más reciente fue hecha en el canal 1
            - ...
        - <29:27>:
            - Reservados
        - <30> (OVERRUN):
            - No es importante si no estamos en modo BURST
            - 0 = El resultado almacenado en RESULT se leyó
            - 1 = El resultado de una conversión (o más) se perdió y fue sobrescrito
        - <31> (DONE):
            - 0 = La conversión aún no finalizó
            - 1 = La conversión finalizó
    - ADINTEN:
        - Controla las interrupciones del ADC
        - <7:0> (ADINTENX):
            - 0 = Deshabilita la interrupción por pin X del ADC cuando se completa una conversión
            - 1 = Habilita la interrupción por pin X del ADC cuando se completa una conversión
        - <8> (ADGINTEN)
            - 0 = Sólo los canales individualmente del ADC van a generar interrupciones
            - 1 = Sólo la bandera global DONE podrá interrumpir
        - <31:17>:
            - Reservados
    - ADDR0-7:
        - Cada registro corresponde a un pin analógico del ADC
        - Almacenan la última conversión del pin correspondiente
        - <3:0>:
            - Reservados
        - <15:4> (RESULT):
            - Resultado de la conversión
        - <29:16>:
            - Reservados
        - <30> (OVERRUN):
            - Mismo funcionamiento que OVERRUN en ADGDR
        - <31> (DONE):
            - Mismo funcionamiento que DONE en ADGDR
    - ADSTAT:
        - Permite ver el estado de los canales del ADC
        - Muestra las flags done y overrun de cada ADDRX
    - ADTRM:
        - Configura el arranque del ADC
        - Este registro no lo vemos

$$pclk = cclk/div_perif$$

$$clock = pclk/div_adc$$

$$f_muest = clock/ciclos$$

$$t_conv = 1/f_muest$$