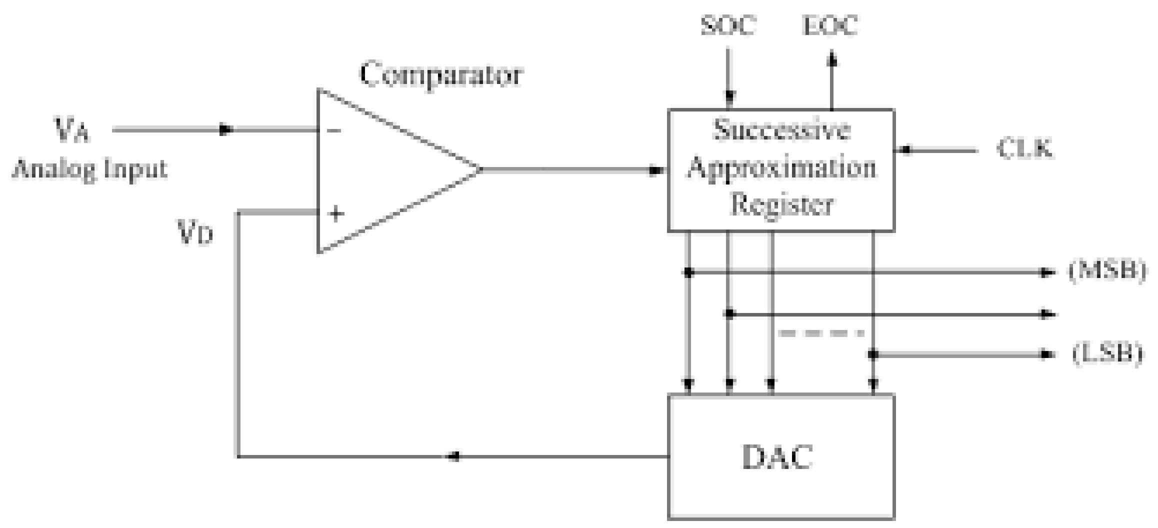
ANALOG to DIGITAL CONVERTER(ADC)

ADC IN LPC1768

- ADC uses Successive Approximation method for analog to digital conversion



ADC of LPC 1768

- Commonly used range 0 to 3.3V
- 12-bit ADC
- Maximum of 8 multiplexed analog inputs are supported

- Smallest voltage that cause variation in digital output is referred as **resolution of ADC** and is (Vrefp-Vrefn)/2^N, Ex: 3.3/2^12 = 0.805 mV

Vanalog= (VREFP / 2^N) (Decimal Equivalent of Digital value(in binary))

Pin CND	Pin LPC1768	Description
1	9	P0.23/AD0.0/I2SRX_CLK/CAP3.0
2	8	P0.24/AD0.1/I2SRX_WS/CAP3.1
3	7	P0.25/AD0.2/I2SRX_SDA/TXD3
4	6	P0.26/AD0.3/AOUT/RXD3
5	25	P0.27/SDA0/USB/SDA
6	24	P0.28/SCL0/USB_SCL
7	75	P2.0/PWM1.1/TXD1
8	74	P2.1/PWM1.2/RXD1
9	-	No connection
10	-	Ground

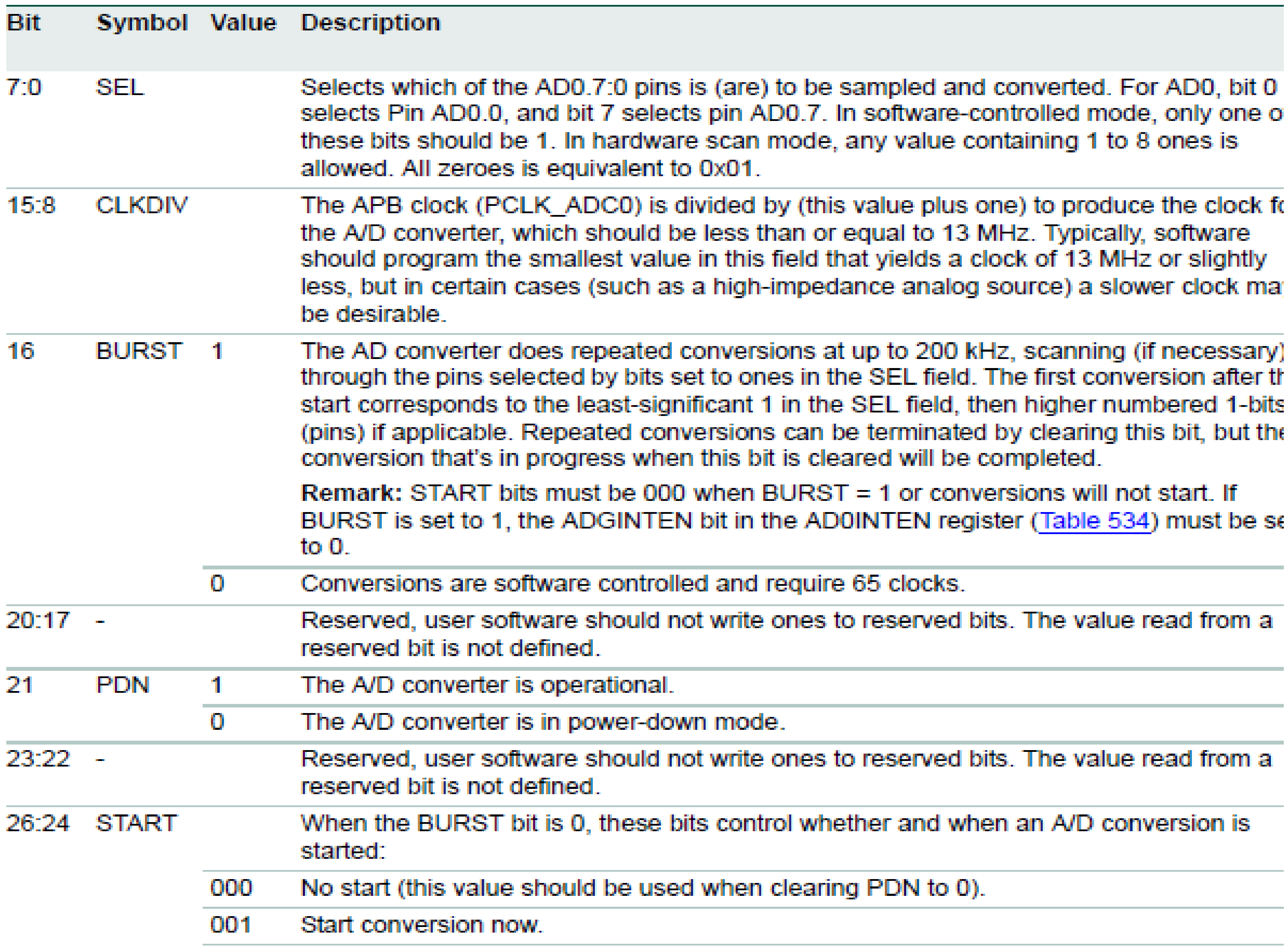
Internally availabale on the kit
AD0.4   P1.30 in fn 3
AD0.5   P1.31 in fn 3

SFRs in ADC

- ADCR : A/D control registers
- ADDR0 – ADDR7 : A/D data register
- ADGDR : A/D global data register
- ADINTEN :A/D interrupt enable register
- ADSTAT : A/D conversion status is updated

Generic Name	Description
ADCR	A/D Control Register. The ADCR register must be written to select the operating mode before A/D conversion can occur.
ADGDR	A/D Global Data Register. This register contains the ADC's DONE bit and the result of the most recent A/D conversion.
ADINTEN	A/D Interrupt Enable Register. This register contains enable bits that allow the DONE flag of each A/D channel to be included or excluded from contributing to the generation of an A/D interrupt.
ADDR0	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0.
ADDR1	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.
ADDR2	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.
ADDR3	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.
ADDR4	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.
ADDR5	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.
ADDR6	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.
ADDR7	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.
ADSTAT	A/D Status Register. This register contains DONE and OVERRUN flags for all of the A/D channels, as well as the A/D interrupt/DMA flag.

ADCR



Bit	Symbol	Value	Description
7:0	SEL		Selects which of the AD0.7:0 pins is (are) to be sampled and converted. For AD0, bit 0 selects Pin AD0.0, and bit 7 selects pin AD0.7. In software-controlled mode, only one o these bits should be 1. In hardware scan mode, any value containing 1 to 8 ones is allowed. All zeroes is equivalent to 0x01.
15:8	CLKDIV		The APB clock (PCLK_ADC0) is divided by (this value plus one) to produce the clock fo the A/D converter, which should be less than or equal to 13 MHz. Typically, software should program the smallest value in this field that yields a clock of 13 MHz or slightly less, but in certain cases (such as a high-impedance analog source) a slower clock ma be desirable.
16	BURST	1	The AD converter does repeated conversions at up to 200 kHz, scanning (if necessary) through the pins selected by bits set to ones in the SEL field. The first conversion after th start corresponds to the least-significant 1 in the SEL field, then higher numbered 1-bits (pins) if applicable. Repeated conversions can be terminated by clearing this bit, but the conversion that's in progress when this bit is cleared will be completed. **Remark:** START bits must be 000 when BURST = 1 or conversions will not start. If BURST is set to 1, the ADGINTEN bit in the AD0INTEN register (Table 534) must be se to 0.
		0	Conversions are software controlled and require 65 clocks.
20:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
21	PDN	1	The A/D converter is operational.
		0	The A/D converter is in power-down mode.
23:22	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
26:24	START		When the BURST bit is 0, these bits control whether and when an A/D conversion is started:
		000	No start (this value should be used when clearing PDN to 0).
		001	Start conversion now.

ADCR initialization

- If input is applied on channel 5 i.e. AD0.5, then ADCR is initialized to

```
LPC_ADC->ADCR = 1<< 5 | 1<<21 | 1<<24;
//21 bit is PDN, 24 is to start the conversion
```

If input is applied to AD0.2 and AD0.5 for burst mode of operation, then LPC_ADC->ADCR =?

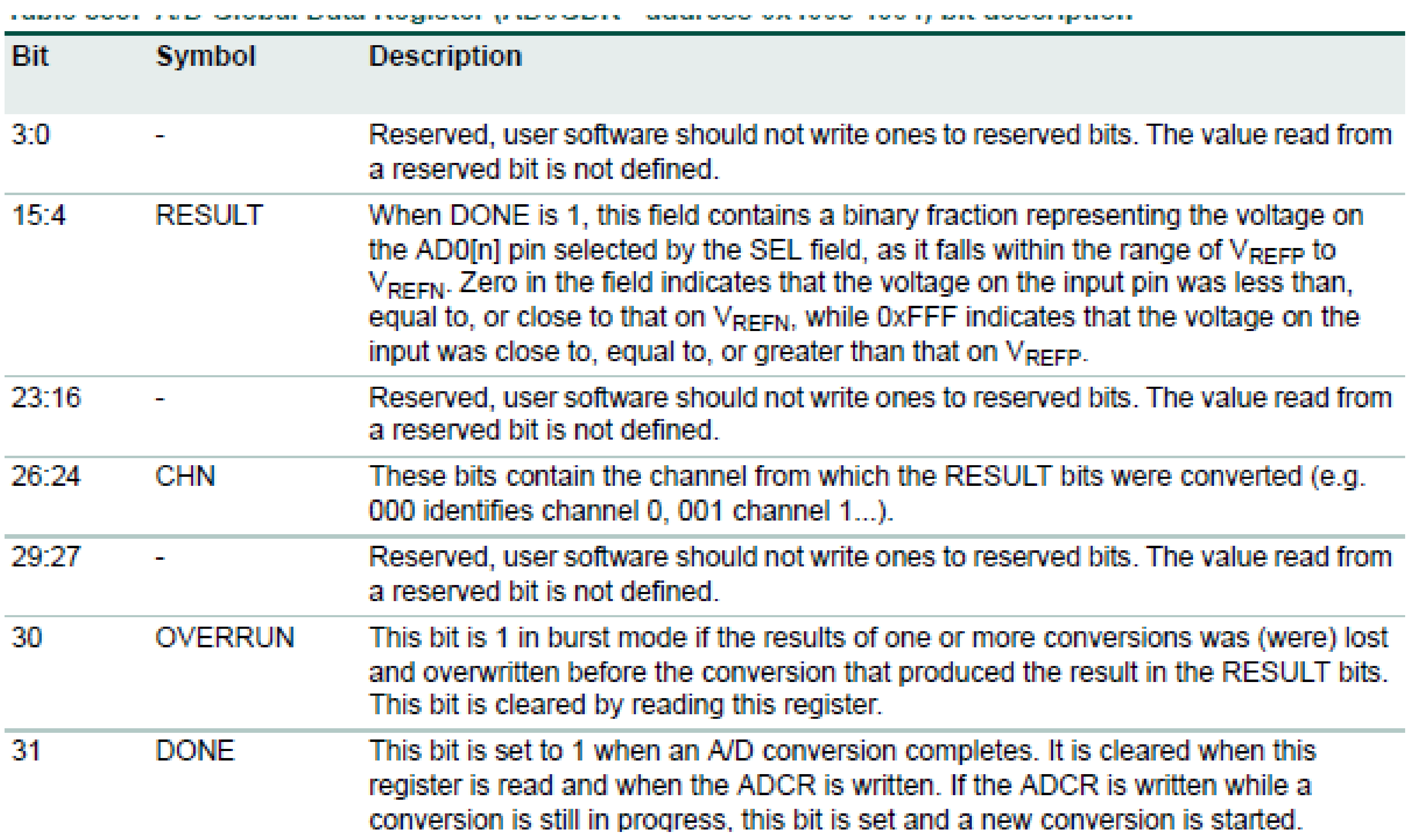
ADGDR

DONE bit (31st bit) is made high after the completion of conversion

A/D global register (15:4) holds the result of the most recent A/D conversion that has completed, and also includes copies of status flags that go with that conversion.

Results of conversion can be read from A/D global register or A/D channel data register.

ADGDR



Bit	Symbol	Description
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
15:4	RESULT	When DONE is 1, this field contains a binary fraction representing the voltage on the AD0[n] pin selected by the SEL field, as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}.
23:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
26:24	CHN	These bits contain the channel from which the RESULT bits were converted (e.g. 000 identifies channel 0, 001 channel 1...).
29:27	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits. This bit is cleared by reading this register.
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read and when the ADCR is written. If the ADCR is written while a conversion is still in progress, this bit is set and a new conversion is started.

ADGDR

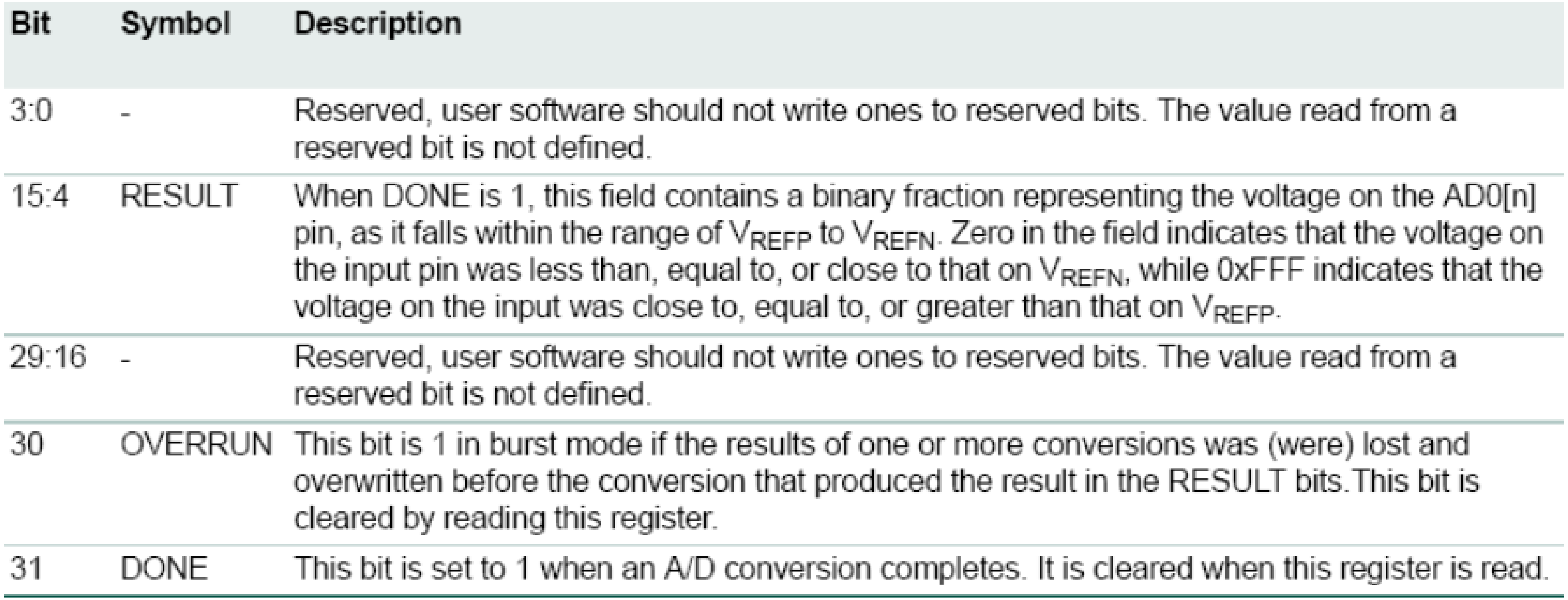
- while(!(LPC_ADC->ADGDR & 0X80000000)); // waiting for done bit to become high

// or while((LPC_ADC->ADGDR &1<< 31)==0);

X = (LPC_ADC->ADGDR>>4)& 0XFFF;

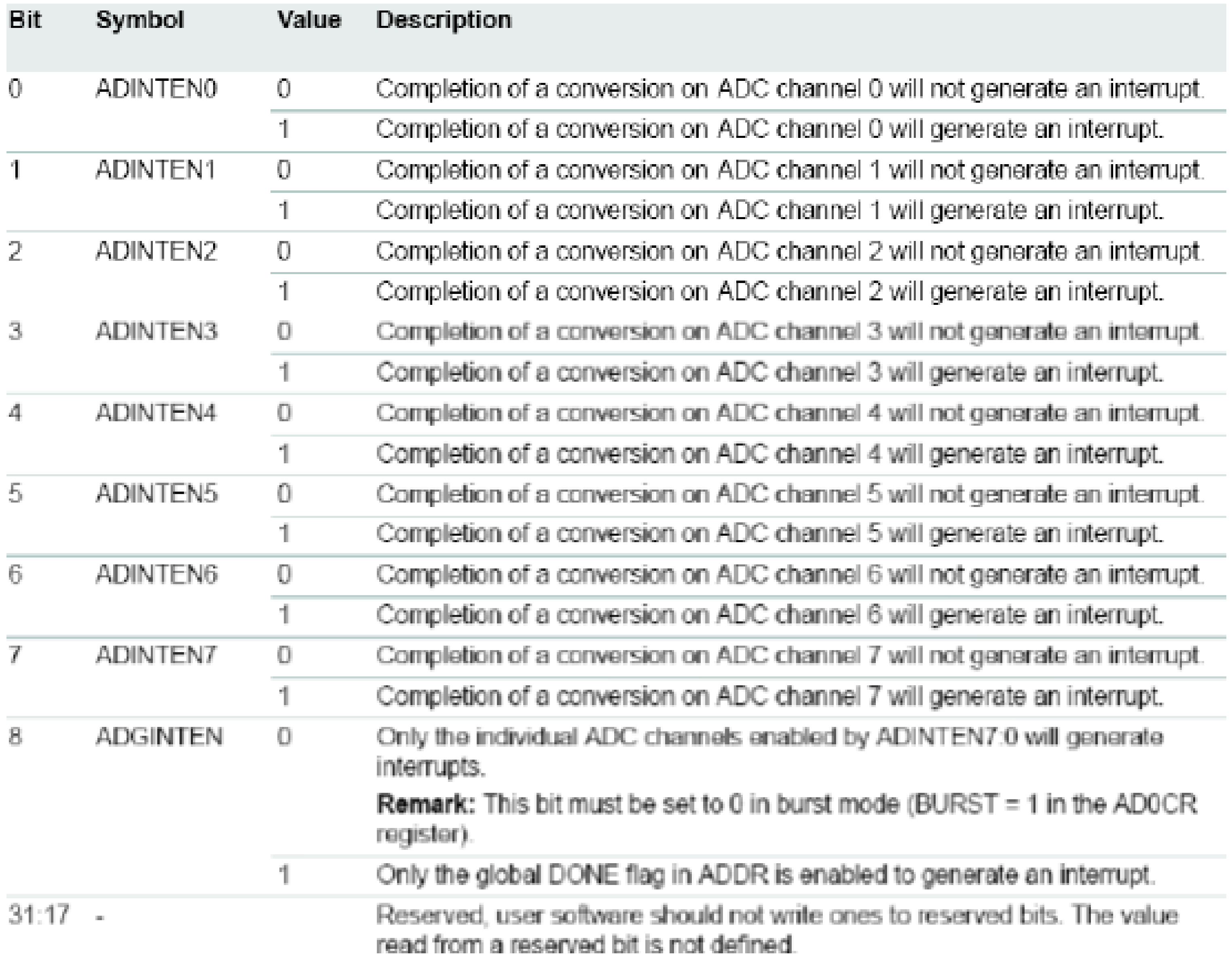
A/D Data Register(ADDR0 TO ADDR7)

- A/D data registers hold the result of last conversion for each channel.
- The completion of A/D conversion is specified by DONE bit 31st



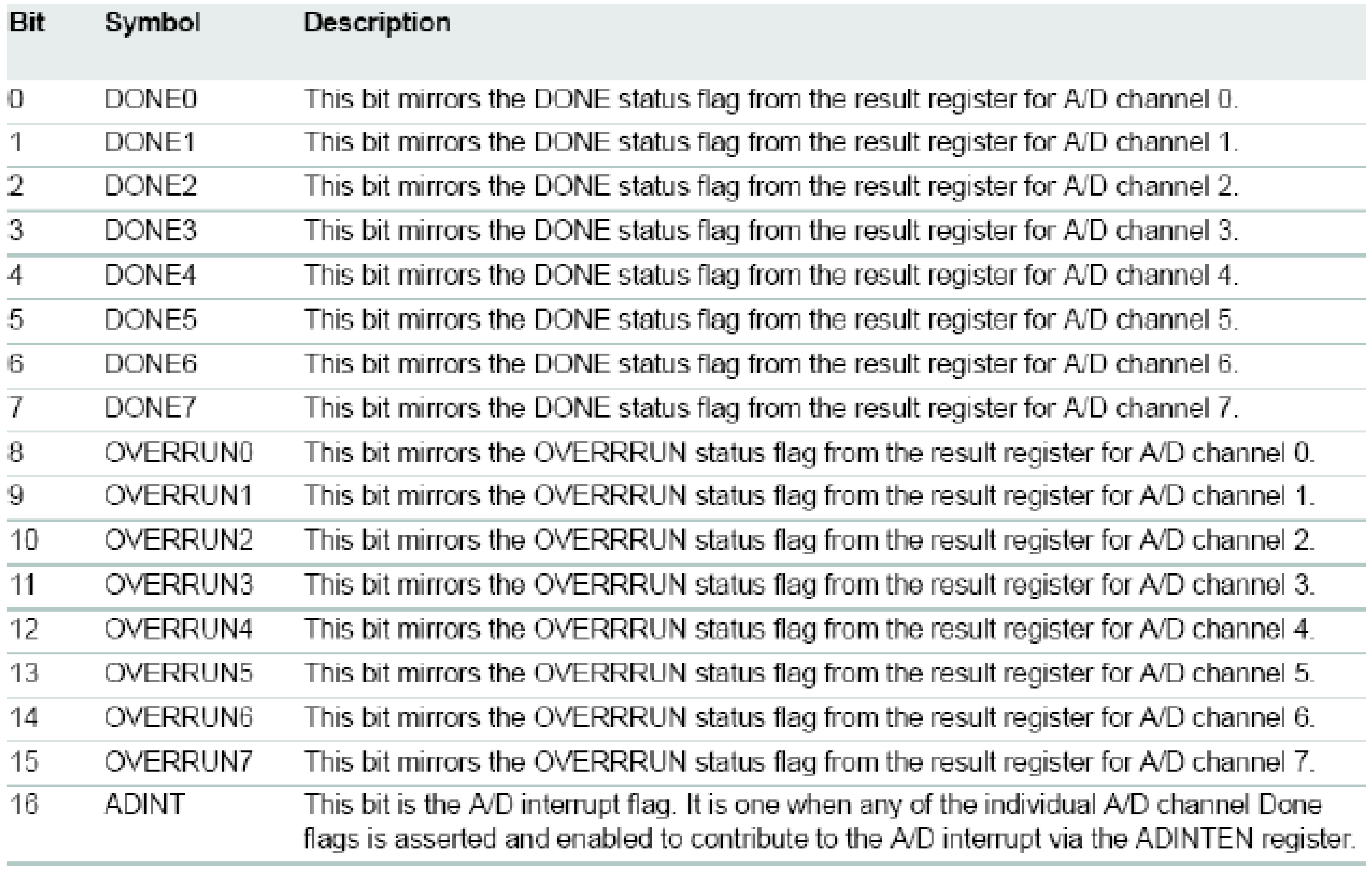
Bit	Symbol	Description
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
15:4	RESULT	When DONE is 1, this field contains a binary fraction representing the voltage on the AD0[n] pin, as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}.
29:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits.This bit is cleared by reading this register.
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read.

ADINTEN



Bit	Symbol	Value	Description
0	ADINTEN0	0	Completion of a conversion on ADC channel 0 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 0 will generate an interrupt.
1	ADINTEN1	0	Completion of a conversion on ADC channel 1 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 1 will generate an interrupt.
2	ADINTEN2	0	Completion of a conversion on ADC channel 2 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 2 will generate an interrupt.
3	ADINTEN3	0	Completion of a conversion on ADC channel 3 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 3 will generate an interrupt.
4	ADINTEN4	0	Completion of a conversion on ADC channel 4 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 4 will generate an interrupt.
5	ADINTEN5	0	Completion of a conversion on ADC channel 5 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 5 will generate an interrupt.
6	ADINTEN6	0	Completion of a conversion on ADC channel 6 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 6 will generate an interrupt.
7	ADINTEN7	0	Completion of a conversion on ADC channel 7 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 7 will generate an interrupt.
8	ADGINTEN	0	Only the individual ADC channels enabled by ADINTEN7:0 will generate interrupts. **Remark:** This bit must be set to 0 in burst mode (BURST = 1 in the AD0CR register).
		1	Only the global DONE flag in ADDR is enabled to generate an interrupt.
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

A/D STATUS REGISTER



Bit	Symbol	Description
0	DONE0	This bit mirrors the DONE status flag from the result register for A/D channel 0.
1	DONE1	This bit mirrors the DONE status flag from the result register for A/D channel 1.
2	DONE2	This bit mirrors the DONE status flag from the result register for A/D channel 2.
3	DONE3	This bit mirrors the DONE status flag from the result register for A/D channel 3.
4	DONE4	This bit mirrors the DONE status flag from the result register for A/D channel 4.
5	DONE5	This bit mirrors the DONE status flag from the result register for A/D channel 5.
6	DONE6	This bit mirrors the DONE status flag from the result register for A/D channel 6.
7	DONE7	This bit mirrors the DONE status flag from the result register for A/D channel 7.
8	OVERRUN0	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 0.
9	OVERRUN1	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 1.
10	OVERRUN2	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 2.
11	OVERRUN3	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 3.
12	OVERRUN4	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 4.
13	OVERRUN5	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 5.
14	OVERRUN6	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 6.
15	OVERRUN7	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 7.
16	ADINT	This bit is the A/D interrupt flag. It is one when any of the individual A/D channel Done flags is asserted and enabled to contribute to the A/D interrupt via the ADINTEN register.

Basic ADC configurations and accessing the digital value p1.31 as AD0.5

```
int main()
{                   SystemInit();
                    SystemCoreClockUpdate ();


                    LPC_SC->PCONP= 1<<12;//To enable ADC
                    LPC_SC->PCONP |= (1<<15); //Power for GPIO block
                    LPC_PINCON->PINSEL3 = 3<<30; //0XC0000000
                    LPC_ADC->ADCR = (1<<5)|(1<<21)|(1<<24);//AD0.5, start conversion and operational
                    While(1)
                    {
                    while(LPC_ADC->ADGDR &1<< 31)==0); //wait till 'done' bit is 1, indicates conversion complete

                    adc_temp = LPC_ADC->ADGDR;
                    adc_temp >>= 4;
                    adc_temp &= 0x00000FFF; //12 bit ADC

                    in_vtg = (((float)adc_temp * (float)Ref_Vtg))/((float)Full_Scale);
                     //calculating input analog voltage
                    sprintf(vtg,"%3.2fV",in_vtg);  //convert the readings into string to display on LCD, defined in stdio.h
                    sprintf(dval,"%x",adc_temp); //digital equivalent
                    //code to display on LCD
                    }
                    }
```

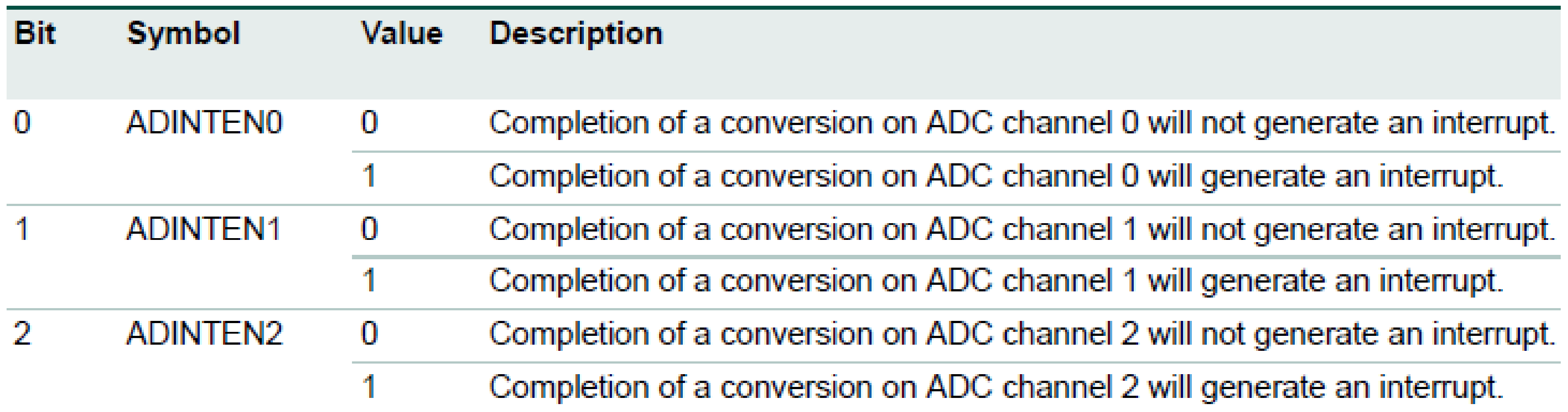
Display the analog input voltage and its digital equivalent

```
include<LPC17xx.h>
 #include<stdio.h>
 #define      Ref_Vtg                    3.300
 #define      Full_Scale    0xFFF//12 bit ADC
int main(void)
{
              unsigned long adc_temp;
              unsigned int i;
              float in_vtg;
              unsigned char vtg[7], dval[7];
              unsigned char Msg3[] = {"ANALOG IP:"};
              unsigned char Msg4[] = {"ADC OUTPUT:"};
              SystemInit();
              SystemCoreClockUpdate();
              lcd_init();//Initialize LCD
               LPC_PINCON->PINSEL3 |= 3<<30;//P1.31 as AD0.5
              LPC_SC->PCONP |= (1<<12);//enable the peripheral ADC
              flag1=0;//Command
              temp1 = 0x80;//Cursor at beginning of first line
              lcd_write();
              flag1=1;//Data
               i =0;
               while (Msg3[i++] != '\0')
                 {
                 temp1 = Msg3[i];
                 lcd_write();//Send data bytes
                 }
```

```
        flag1=0; //Command
            temp1 = 0xC0;//Cursor at beginning of second line
          lcd_write();
          flag1=1;
            i =0;
          while (Msg4[i++] != '\0')
            {
            temp1 = Msg4[i];
            lcd_write();//Send data bytes
             }
while(1)
{
         LPC_ADC->ADCR = (1<<5)|(1<<21)|(1<<24);//ADC0.5, start conversion and operational
           while(((adc_temp=LPC_ADC->ADGDR) & (1<<31)) == 0);
         adc_temp = LPC_ADC->ADGDR;
         adc_temp >>= 4;
         adc_temp &= 0x00000FFF;          //12 bit ADC
         in_vtg = (((float)adc_temp * (float)Ref_Vtg))/((float)Full_Scale);        //calculating input analog voltage
         sprintf(vtg,"%3.2fV",in_vtg);        //convert the readings into string to display on LCD
         sprintf(dval,"%x",adc_temp);
flag1=0;;
temp1 = 0x8A;
lcd_write();
flag1=1;
      i =0;
   while (vtg[i++] != '\0')
     {
      temp1 = vtg[i];
      lcd_write();//Send data bytes
       }
flag1=0;
temp1 = 0xCB;
lcd_write();
flag1=1;
  i =0;
 while (dval[i++] != '\0’)
   {
    temp1 = dval[i];
    lcd_write();//Send data bytes
    }
   for(i=0;i<7;i++)
   vtg[i] = dval[i] = 0;
}
}
```

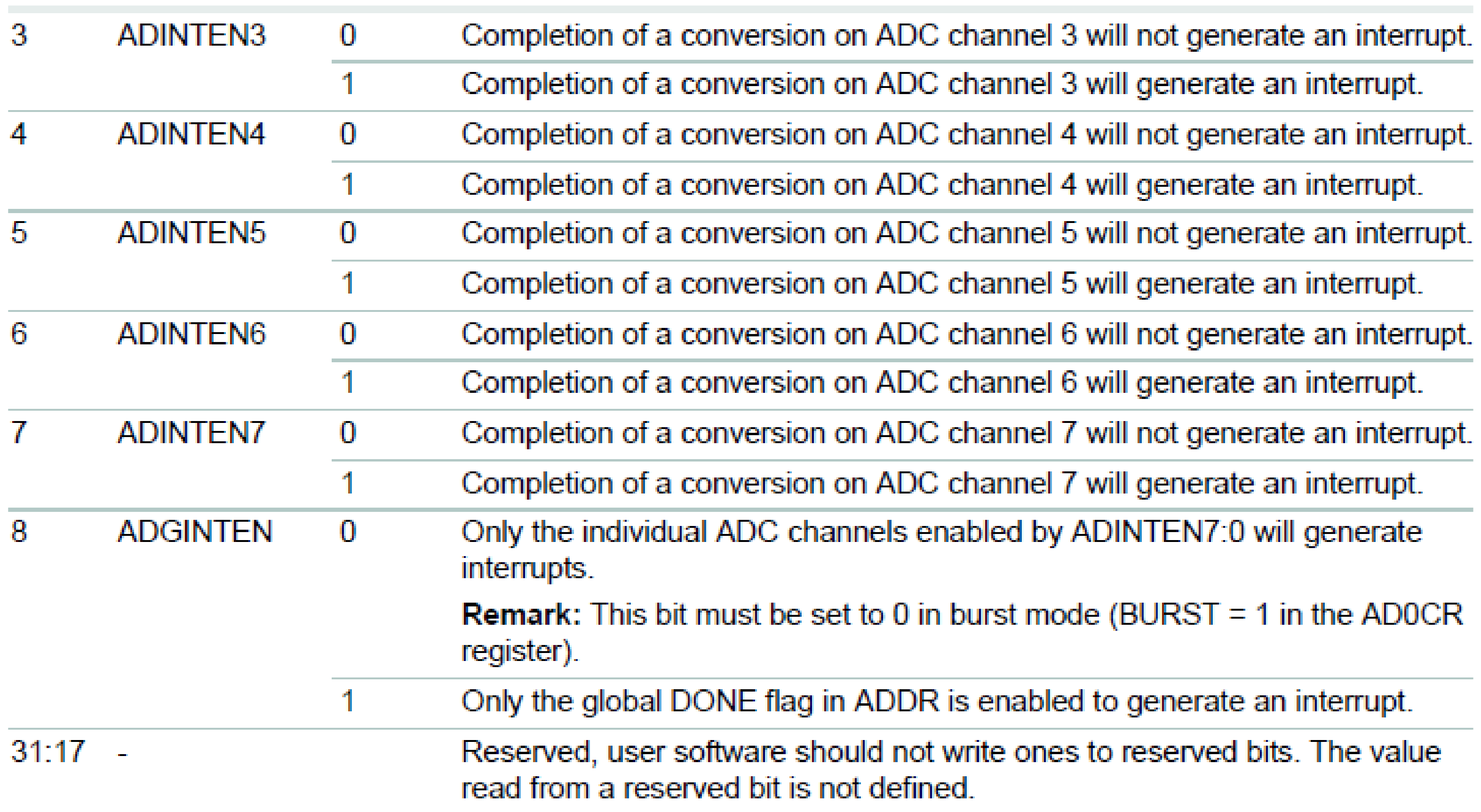
ADC with interrupts

- SFRs:
  - ADINTEN : To enable the interrupts



Bit	Symbol	Value	Description
0	ADINTEN0	0	Completion of a conversion on ADC channel 0 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 0 will generate an interrupt.
1	ADINTEN1	0	Completion of a conversion on ADC channel 1 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 1 will generate an interrupt.
2	ADINTEN2	0	Completion of a conversion on ADC channel 2 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 2 will generate an interrupt.

AD0INTEN CONTD..

- 

3	ADINTEN3	0	Completion of a conversion on ADC channel 3 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 3 will generate an interrupt.
4	ADINTEN4	0	Completion of a conversion on ADC channel 4 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 4 will generate an interrupt.
5	ADINTEN5	0	Completion of a conversion on ADC channel 5 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 5 will generate an interrupt.
6	ADINTEN6	0	Completion of a conversion on ADC channel 6 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 6 will generate an interrupt.
7	ADINTEN7	0	Completion of a conversion on ADC channel 7 will not generate an interrupt.
		1	Completion of a conversion on ADC channel 7 will generate an interrupt.
8	ADGINTEN	0	Only the individual ADC channels enabled by ADINTEN7:0 will generate interrupts. **Remark:** This bit must be set to 0 in burst mode (BURST = 1 in the AD0CR register).
		1	Only the global DONE flag in ADDR is enabled to generate an interrupt.
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

Find the difference between the digital value AD0.4 and AD0.5

```
int main()
{
        SystemInit();
        SystemCoreClockUpdate();
        LPC_PINCON->PINSEL3 = 0xF0000000; //p1.30 as AD0.4
        LPC_SC_PCONP | =1<<12;
        LPC_ADC->ADCR = 1<<4 | 1<<5 | 1<<16 |1<<21;
        LPC_ADC_ADINTEN = 1<<4 | 1<<5;
        NVIC_EnableIRQ(ADC_IRQn);
        while(1);
}
```

Find the difference between the digital value AD0.4 and AD0.5 contd..

```
Void ADC_IRQHandler()
{               int x;
                int channel, temp1, temp2;
                X = LPC_ADC->ADSTAT;
                X=X & 3<<4;
                if (X ==1<<4)
                {
                                temp1 = LPC_ADC->ADDR4>>4;
                }
                elseif (X ==1<<5)
                                { temp2 = LPC_ADC->ADDR5>>4;
                }
                //COMPLETE THE CODE TO DISPLAY THE DIFFERENCE ON MULTIPLEXED SEGMENTS
}
```