Digital to Analog Conversion using ARM Processor

P.S.S SUSHAMA, C.NAGARAJA, K. NAGABHUSHAN RAJU and K. MALAKONDAIAH

ABSTRACT--- Embedded Processor or controller is a logic device. The processor, RAM, ROM,I/O ports, timers/counters etc are all embedded together on one chip with many built in functions. Its advanced architecture reduces the external hardware to a greater extent. Embedded processor/ controller process only digital signals that are binary and discontinuous. Hence, a digital signal needs to be translated in to an analog signal to represent a physical quantity. The present paper deals with digital to analog conversion using ARM Processor LPC2366.

1. Introduction

Embedded processor/controller is a logic device. The processor, RAM, ROM, I/O ports, timers/counters etc are all embedded together on one chip with many built in functions[1-5]. In the real-world, physical quantities like temperature, presseure,etc. are continious and these are represented by equivalent electrical quantities called analog signals. But the digital processor understands only binary language. So it is necessary to convert analog signal to digital form before accessing of data by any processor/controller . Similaly after processing of data which is in digital form is to be again converted in to equivalent analog signal.The converted analog signal can be used as a dynamic signal for feedback control applications.

Most modern audio signals are stored in digital form (for example MP3s and CDs) and in order to be heard through speakers they must be converted into an analog signal. DACs are therefore found in CD players, digital music players, and PC sound cards. Specialist standalone DACs can also be found in high-end hi-fi systems. These normally take the digital output of a compatible CD player or dedicated transport and convert the signal into an analog line-level output that can then be fed into an amplifier to drive speakers. Similar digital-to-analog converters can be found in digital speakers such as USB speakers and in sound cards. In VOIP (Voice over IP) phone, data transmission over the internet is done digitally. So in order for voice to be transmitted it must be converted to digital using an analog-to-digital converter and be converted into analog again using a DAC so that the voice it can be heard on the other end.A device that is distantly related to the DAC is the digitally controlled potentiometer, used to control an analog signal digitally.Hence, the need to convert the digital signals in to analog .[6]

2. Experimental

In earlier stages when we want to get the data in the analog or digital form we need to interface externally with the microcontroller(like 8051/52 etc…) Now a days in ARM controller there is a provision of inbuilt DAC or ADC within the chip in addition to the general features. This will reduce the external component interface and programming complexity. No doubt, there are saveral microcontrollers with which digital to analog conversion can be achieved with proper interfacing of the DAC chips externally. But the efforts to make use of the internal DAC prohibition of certain advanced microcontrollers like ARM processors are rather scarce. Hence, in the present study a attempt is made to implement the digital to analog conversion on making use of the internal DAC of the ARM processor.

Hardware details :

The ARM Processor has a built in 10-bit DAC of register string architecture [7-9]. The maximum output value of the DAC is Vi (VREF). The voltage on this pin is VALUE/1024 × VREF. It also has power-down mode. It has buffered output with a selectable output drive. The DAC allows the LPC2364/66/68 generated a variable analog output corresponding to the digital input stored in its read/write register[10-12] (DACR).

The hardware part of the ARM controller LPC2366 used to implement the digital to analog conversion is shown in Fig.1. The analog output can be obtained through a port pin which can be enabled using a pin selection register as shown in the Fig.1. The output of DAC is observed on a cathode ray oscilloscope.

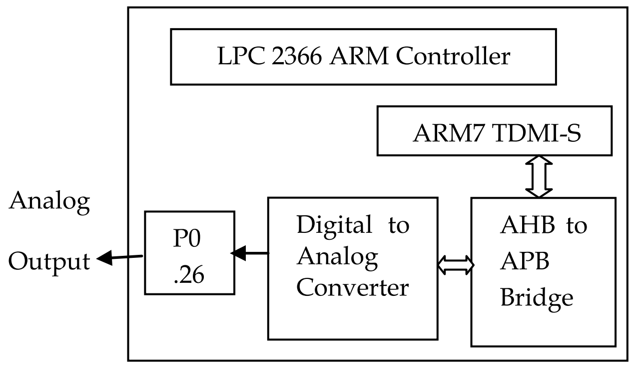

Figure 1. Digital to Analog conversion using ARM processorLPC2366.

Software description:

Algorithm for digital to analog conversion is as follows.

1.Slect Port 0 pin (P0.26) to present DAC output using PINSEL1 register .
2.Load the digital data in an array (0x00,0x0F,0x1F,0x2F,0x3F,0x4F,0x5F, 0x6F,0x7F,0x8F,0x9F,0xAF,0xBF,0xCF, 0xDF,0xEF,0xFF,0x100,0x150,0x200, 0x250,0x300,0x350,0x399).
3. Set counter value to 0.
4. Send the array value to DACR (read/write register) of ARM controller.
5. Observe the output at P0.26 pin.
6. Call Delay.
7. Increment counter.
8. Repeate step 4 to 6 till counter value reaches 25.
9. Repeate from step 3 to 8.

Some of the registers of LPC 2366 are used in the program as mentioned in the User Manual.[13]

The program, in detail, written in Embedded C (Keil IDE µvision V4.00)[14] follows.

Program for Digital to Analog conversion:

```
// ----- delay program ---//
void wait(void)
{
unsigned int delay;
for(delay=0;delay<0x60000;delay++);
for(delay=0;delay<0x60000;delay++);
}
```

```
//--------Main program ----------//

#include<lpc23xx.h>
unsigned int i = 0;
unsignedintdatatable[25]={0x00,0x0F,0x1F,0x2F,0x3F,0x4F,
0x5F,0x6F,0x7F,0x8F,0x9F,0xAF,0xBF,0xCF,0xDF,0xEF,
0xFF,0x100,0x150,0x200,0x250,0x300,0x350,0x400};

 int main (void)
 {
   PINSEL1 = 0x00200000 ;
   while (1)
    {
     for (i=0; i<25; i++)
     {

       DACR = (datatable[i]<<6 );

       wait(); wait(); wait();wait();    wait();

       }}}
```

The execution of the above program will give the DAC output from zero volts to V_{ref} corresponding to the data given in array. The results are recorded and presented in Table1.

S.No.	Digital Input In Hex	Analog Output In Volts
1	00	0.00
2	0F	0.0483
3	1F	0.0999
4	2F	0.1515
5	3F	0.2030
6	4F	0.2546
7	5F	0.3062
8	6F	0.3577
9	7F	0.4093
10	8F	0.4608
11	9F	0.5124
12	AF	0.5640
13	BF	0.6155
14	CF	0.6671
15	DF	0.7181
16	EF	0.7702
17	FF	0.8218
18	100	0.8250
19	150	1.0828
20	200	1.6500
21	250	1.9078
22	300	2.4750
23	350	2.7328
24	399	3.2798

To generate wave forms using DAC:

a) Square wave:

Algorithm for generating the Square wave:

1. Select port0 pin (P0.26) to present DAC output using PINSEL1 register.
2. Load the digital value with 00H.
3. Send the digital value to DACR register.
4. Observe the output at P0.26.
5. Call delay.
6. Load the digital value with 399H.
7. Call delay.
8. Repeat from Step2 to 8.
9.

Some of the registers of LPC 2366 are used in the program as mentioned in the User manual.[13]

The program, in detail, written in Embedded C (Keil IDE µvision V4.00)[14] follows.

Program for generating a square wave:

```
 #include <LPC23xx.H>
// ----------Delay Program------//
Void wait (void)
{
Unsigned int delay;
For (delay=0; delay<0x600000; delay++);
}
//-------Main program---//

Int main ()
{
Unsigned int a,b;
PINSEL1=0x00200000;
While (1)
{
a=0x00;
DACR=a<<6;
Wait ();
Wait ();
Wait ();
b=0x399;
DACR=b<<6;
Wait ();
Wait ()
}
}
```

The execution of program will give a square wave at the DAC output. Observe the waveform with C.R.O. Changing the duration of the delay, the frequency of the wave can be changed.

3.2798

b) Ramp wave :

Algorithim for generating the Ramp wave.

1) Set the DAC output pin as Port 0 (P0.26) pin. by PINSEL1 register .
2). To generate Ramp wave take values in a for loop from 0 and end with 1024 to get the ramp wave.
3). Send the data to the DACR to get the output.
4). See the result on P0.26 pin and observe the wave forms in a CRO.

Some of the registers of LPC 2366 are used in the program as mentioned in the User Manual .[13]

The program, in detail, written in Embedded C (Keil IDE µvision V4.00)[14] follows.

Program to generate Ramp wave:

```
//--------main program------------------//

#include<lpc23xx.h>
 unsigned int i = 0;
 int main (void)
   {
     PINSEL1 = 0x02000000 ;
     while (1)
   {
   for(;;)
   {
for(i = 0 ; i < 1024 ; i++)     // 00 to ff to get the ramp wave
DACR =i << 6 ;      // senidng the values to the DAC register
```

The excecution of the program will give a Ramp wave at the DAC output. Observe the wave form with C.R.O. Changing the duration of the delay, the frequency of the wave can be changed.

c) Sine wave :

Algorithim for generating the sine wave:

1. Set the DAC output pin as Port 0 (PO.26) pin. Put it in PINSEL1 register .
2. To generate sine wave take a take an array and prepare the sine tables to generate sine wave .
3. An array of sine values are taken.
4. Send the data to the DACR i.e DAC register to get the output.
5. See the result on Po.26 pin and observe the wave forms in a CRO.

wait();

Some of the registers of LPC 2366 are used in the program as mentioned in the User manual.[13]

The program, in detail, written in Embedded C (Keil IDE µvision V4.00)[14] follows.

Program to generate sine wave:

```
#include<lpc23xx.h>
unsigned int i = 0;
unsigned int SinTable[45] ={410, 467, 523, 576, 627, 673, 714,
749, 778, 799,813, 819, 817, 807, 789, 764, 732, 694, 650, 602,
550, 495,438, 381, 324, 270, 217, 169, 125, 87, 55, 30, 12, 2,  0,
6, 20, 41, 70, 105, 146, 193, 243, 297, 353};
  unsigned int i = 0 ;
  int main (void)
   {
PINSEL1 = 0x00200000 ;// to use P0.25 as DAC output pin.
While(1)
 {
for (i = 0x00; i<0x2D; i++)     // 45 values are taken in a
loop.
 DACR = (SinTable[i] << 6);  // sine values sending to the
DACR regsiter.
      }

}

}
```

The excecution of the program will give a sine wave at the DAC output. Observe the wave form with C.R.O. Changing the duration of the delay, the frequency of the wave can be changed.

Conclusions

The hardware and software features of Digital to Analog conversion using ARM processor LPC2366 are discussed in the present study. These studies will help a lot in using the ARM processor for any data acquisition applications.

Acknowledgements

The authors are thankful to the University Grants commission, New Delhi for providing financial assistance through UGC Major Research Project.

References

[1] Embedded Controllers and Processors, Intel (1992).
[2] Embedded Controllers Applications, Intel (1991).
[3] Advanced Micro Devices, Microcontroller Hand – book (1988).
[4] Elector India, May 1992- December 1995.
[5] S. Bhasker Reddy, K. Malakondaiah and S. Raja Ratnam, Interfacing Digital-to-Analog converters with a Microcontroller, Physics Education, October-December, p. 233-244(1997).
[6] http://en.wikipedia.org/wiki/Digital-to-analog_converter
[7] http://en.cnki.com.cn/Article_en/CJFDTOTAL-LJMG200712013.htm
[8] http://www.eetimes.com/design/analog design/4009983/Understand-the-differences-between-R2R-and-String-DAC-architectures/
[9] http://inst.eecs.berkeley.edu/~ee247/fa04/fa04/lectures/L15_f04.pdf.
[10] Andrew N.Sloss, Dominic Symes, Chris Wright, ARM System Developer's Guide Design and Optimizing System Software, Morgan Kaufmann Publishers(2011).
[11] David Seal. Addison- Wesley, ARM Architecture Reference manual, Second Edition (2011), the definitive reference for the ARM architecture definition.
[12] Arm System –on-chip Architecture, second Edition, and Published 2000, by Steve Furber. Addision-Wesley. Covers the hardware aspects of ARM processors and SOC design.
[13] www.keil.com/dd/docs/datashts/philips/lpc23xx_um.pdf
[14] http://www.keil.com/arm/mdk.asp