Document location: http://antiradio.narod.ru/psoc/projects/psoc_analog/psoc_analog_en.pdf
Russian version: http://antiradio.narod.ru/psoc/projects/psoc_analog/psoc_analog.pdf
Created: 12/26/2004. Updated: 06/27/2008.

How fast the analogue part of PSoC is working?

(researching of CY8C27443)

For such high speed applications like the TV signal generating, the quality of multilevel output pulses is interesting aspect of PSoC. The DAC6 User Module has the declared update rate 250 kSamples/sec (4µs per sample) which allow generating only 16 elements per PAL video string (64µs). What the ways to improve the productivity?

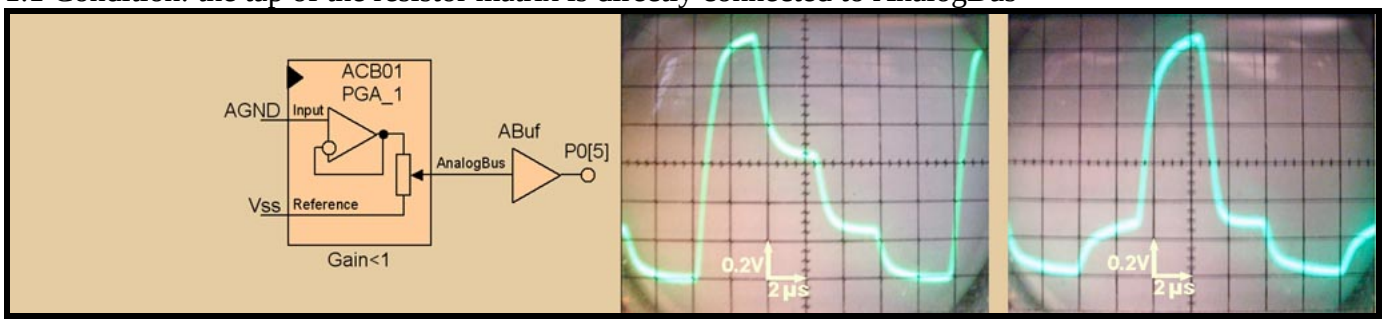
The CY8C27443 chip has the following specifications at **Ucc=5V**, **A_Buff_Power=High** and **Op-Amp Bias=High (**According Final Data Sheet **No 38-12012 Rev. *G**):

	Operational Amplifiers	**Output Buffers**
Rising Slew Rate (10 pF load, Unity Gain)	8.5 V/µs	0.9 V/µs
Falling Slew Rate (10 pF load, Unity Gain)	6 V/µs	0.9 V/µs
Gain Bandwidth Product **Small Signal Bandwidth** (20mVpp, 3dB, 100pF)	8 MHz 1 MHz	

The possibilities of Op-Amps (which are the parts of CT and SC blocks) are much better than the Output Buffers ones.

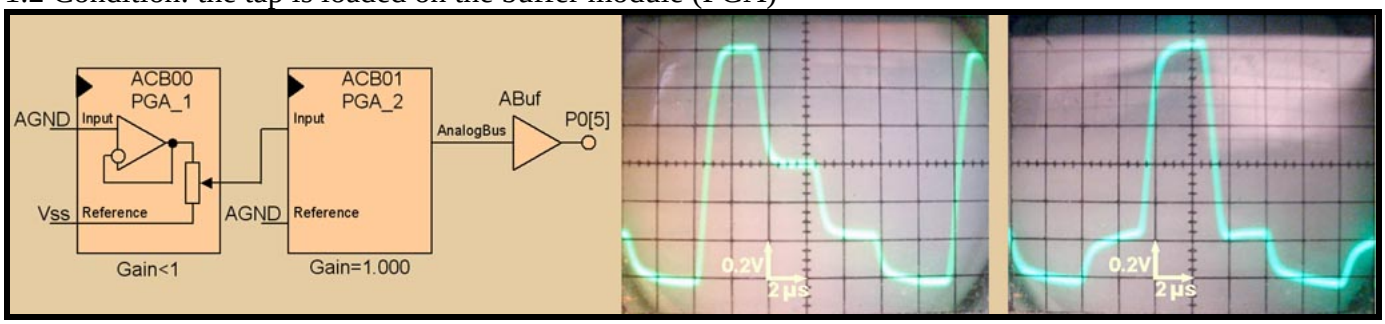
Method 1: The switching of resistor tap in PGA

1.1 Condition: the tap of the resistor matrix is directly connected to AnalogBus



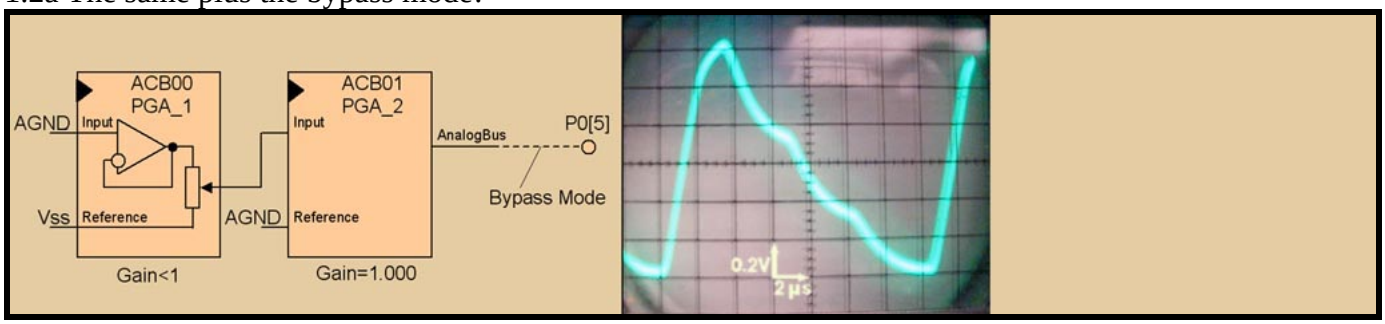
Ref Mux=(Vdd/2)+/-(Vdd/2) Code 1 in Appendix Code 2 in Appendix

1.2 Condition: the tap is loaded on the buffer module (PGA)



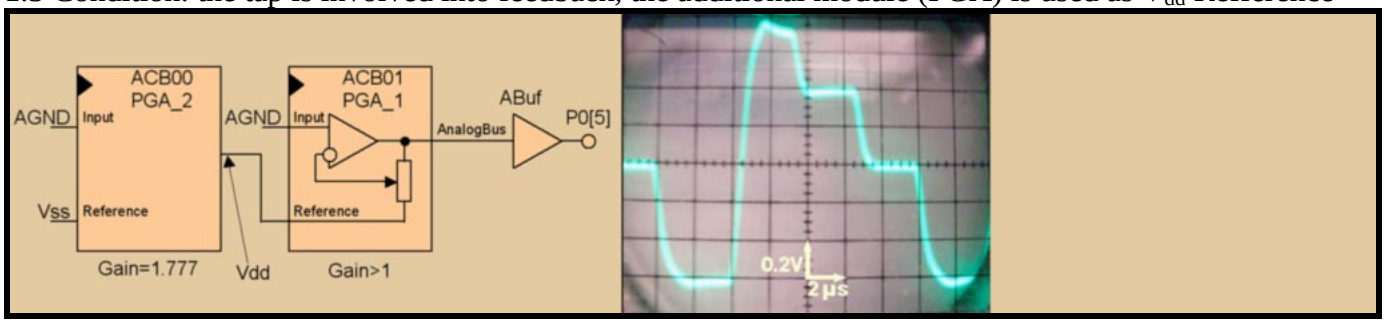
Ref Mux=(Vdd/2)+/-(Vdd/2) Code 3 in Appendix Code 4 in Appendix

1.2a The same plus the bypass mode!



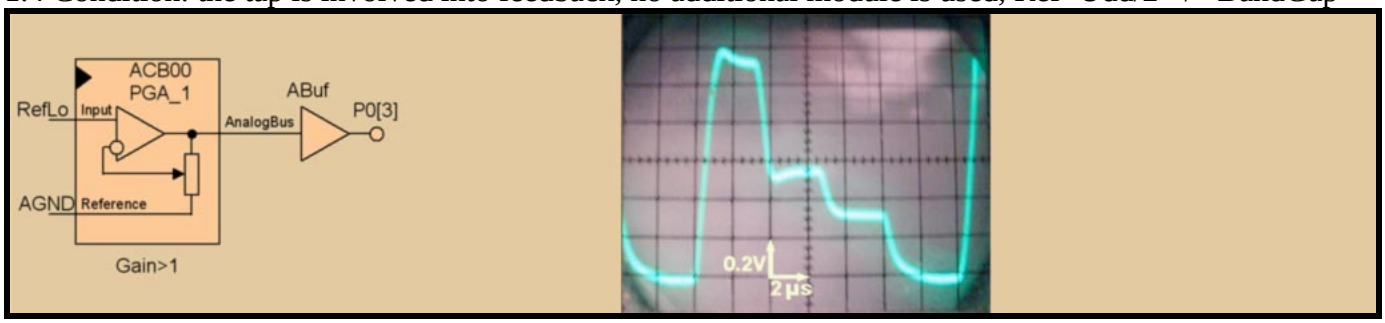
Ref Mux=(Vdd/2)+/-(Vdd/2) Code 5 in Appendix

1.3 Condition: the tap is involved into feedback, the additional module (PGA) is used as V_{dd} Refference



Ref Mux=(Vdd/2)+/-(Vdd/2) Code 6 in Appendix

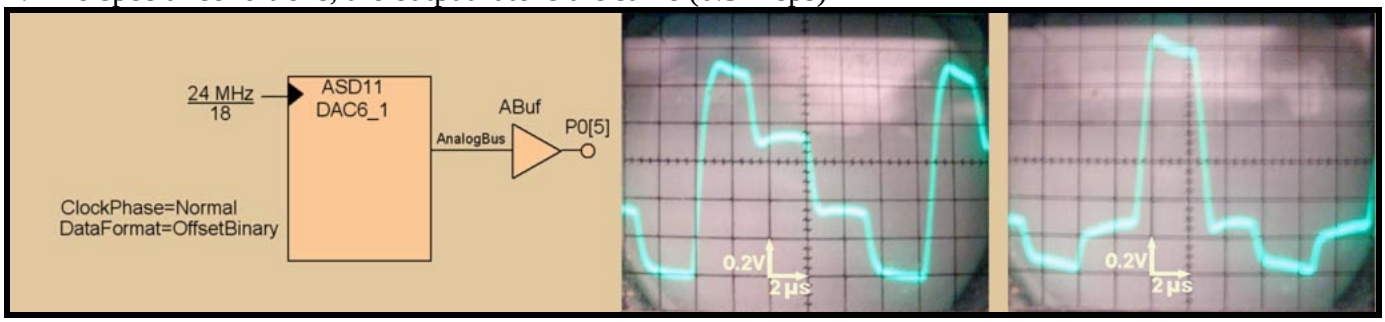
1.4 Condition: the tap is involved into feedback, no additional module is used, Ref=Udd/2 +/– BandGap



Ref Mux=(Vdd/2)+/-(BandGap) Code 7 in Appendix

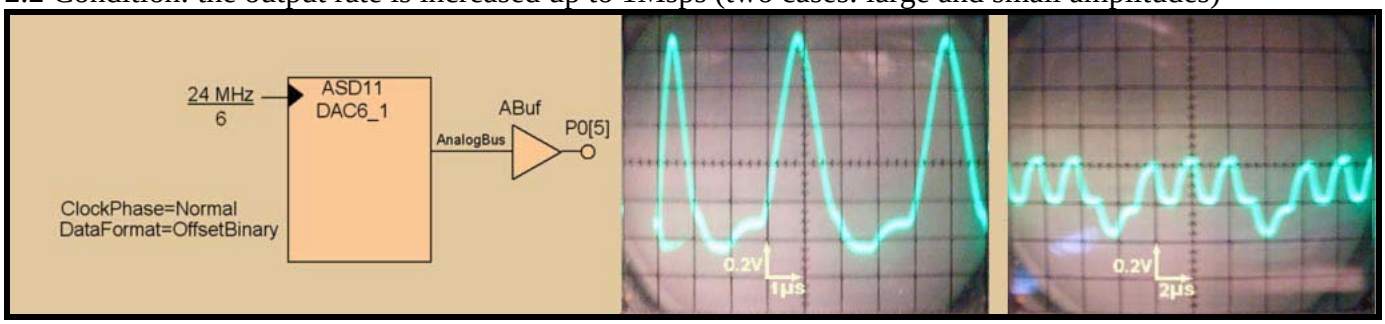
Method 2: DAC6 direct register programming

2.1 No special conditions, the output rate is the same (0.3 Msps)



Ref Mux=(Vdd/2)+/-(Vdd/2) Code 8 in Appendix Code 9 in Appendix

2.2 Condition: the output rate is increased up to 1Msps (two cases: large and small amplitudes)



Ref Mux=(Vdd/2)+/-(Vdd/2) Code 10 in Appendix Code 11 in Appendix

Conclusion

The output rate is limited by the Analog Output Buffer. No visible difference between any presented variations, excluding the case 1.1 where the tap is loaded on capacity of Analog Bus. The bypass mode is dramatically Reduce the slew rates.

Appendix – Code Examples

Code 1

```
;PGA tap directly to Abus
;CPU_Clock=6_MHz


_main:
mov A,3
lcall PGA_1_Start




loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x70    ;8/16 (devider)
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x30    ;4/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x00    ;1/16
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 3

```
;PGA tap to Abus via buff PGA
;CPU_Clock=6_MHz


_main:
mov A,3
lcall PGA_1_Start
mov A,3
lcall PGA_2_Start




loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x70    ;8/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x30    ;4/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x00    ;1/16
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 5

```
;PGA tap to Abus via PGA,
;bypass mode
;CPU_Clock=6_MHz
;Bypass mode
_main:
M8C_SetBank1      ;turn on
or reg[ABF_CR0],2;bypass
M8C_SetBank0      ;mode
mov A,3
lcall PGA_1_Start
mov A,3
lcall PGA_2_Start


loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x70    ;8/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x30    ;4/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x00    ;1/16
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 6

```
;PGA tap to feedback
;Additional PGA as a Ref
;CPU_Clock=6_MHz
_main:
mov A,3
lcall PGA_1_Start
mov A,3
lcall PGA_2_Start


loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0xA0    ;33/48
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x90    ;30/48
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x80    ;27/48
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x70    ;24/48
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 7

```
;PGA tap to feedback
;BandGap as a Reference
;CPU_Clock=6_MHz
_main:
mov A,3
lcall PGA_1_Start
mov A,3
lcall PGA_2_Start


loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0xF0    ;48/48
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0xA0    ;36/48
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x90    ;30/48
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x80    ;27/48
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 8

```
;DAC6
;CPU_Clock=12_MHz
;Column_Clock=24MHz/18
_main:
mov A,3
lcall DAC6_1_Start
mov A,0
lcall DAC6_1_WriteBlind
M8C_Stall


loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,20      ;
mov reg[DAC6_1_CR0],A

mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,24      ;
mov reg[DAC6_1_CR0],A

mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,28      ;
mov reg[DAC6_1_CR0],A

mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,31      ;
mov reg[DAC6_1_CR0],A

jmp loop1
```

Code 2

```
;PGA tap directly to Abus,
;3-levels signal
;CPU_Clock=6_MHz

_main:
mov A,3
lcall PGA_1_Start



loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x70    ;8/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x00    ;1/16
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 4

```
;PGA tap to bus via buff PGA,
;3-levels signal
;CPU_Clock=6_MHz

_main:
mov A,3
lcall PGA_1_Start
mov A,3
lcall PGA_2_Start

loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x70    ;8/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x10    ;2/16
mov reg[PGA_1_GAIN_CR0],A

mov A,reg[PGA_1_GAIN_CR0]
and A,0x0F
or  A,0x00    ;1/16
mov reg[PGA_1_GAIN_CR0],A

jmp loop1
```

Code 9

```
;DAC6, 3-levels signal
;CPU_Clock=12_MHz
;Column_Clock=24 MHz/18
_main:
mov A,3
lcall DAC6_1_Start
mov A,0
lcall DAC6_1_WriteBlind
M8C_Stall

loop1:
;6+4+4+5=19 cycles per sample
mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,29      ;
mov reg[DAC6_1_CR0],A

mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,19      ;
mov reg[DAC6_1_CR0],A

mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,29      ;
mov reg[DAC6_1_CR0],A

mov A,reg[DAC6_1_CR0]
and A,0xE0
or  A,31      ;
mov reg[DAC6_1_CR0],A

jmp loop1
```

Code 10

```
;DAC6, 3-levels signal 1 MS/s
;CPU_Clock=24_MHz
;Column_Clock=4 MHz


_main:
mov A,3
lcall DAC6_1_Start
mov A,0
lcall DAC6_1_WriteBlind
M8C_Stall

loop1:
;9 cycles per single change (9/24us)
xor reg[DAC6_1_CR0],0b00000010;=29
xor reg[DAC6_1_CR0],0b00001110;=19
xor reg[DAC6_1_CR0],0b00001110;=29
or  reg[DAC6_1_CR0],0b00011111;=31

jmp loop1
;9x4+5=41
```

Code 11

```
;DAC6, small signal, 1 MS/s
;CPU_Clock=24_MHz
;Column_Clock=4 MHz
_main:
mov A,3
lcall DAC6_1_Start
mov A,0
lcall DAC6_1_WriteBlind
M8C_Stall

loop1:
;9 cycles per single change (9/24us)
nop
xor reg[DAC6_1_CR0],0b00000010;=29
xor reg[DAC6_1_CR0],0b00000110;=27
xor reg[DAC6_1_CR0],0b00000110;=29
xor reg[DAC6_1_CR0],0b00000110;=27
xor reg[DAC6_1_CR0],0b00000110;=29
xor reg[DAC6_1_CR0],0b00000110;=27
xor reg[DAC6_1_CR0],0b00000110;=29
or  reg[DAC6_1_CR0],0b00011111;=31

jmp loop1
;9x4+5=41
```