

Насколько быстрая аналоговая часть у PSoC?

(небольшое исследование чипа CY8C27443)

Для использования в таких приложениях, как генерирование телевизионных сигналов, было бы интересно узнать, на что способны микросхемы PSoC. Заявленное быстродействие модуля пользователя DAC6 – 250 килоотсчетов/сек (4 мкс на один отсчет), что позволяет получить в стандартной видеостроке (длительностью 64 мкс) только 16 элементов. Есть ли способы улучшить этот параметр?

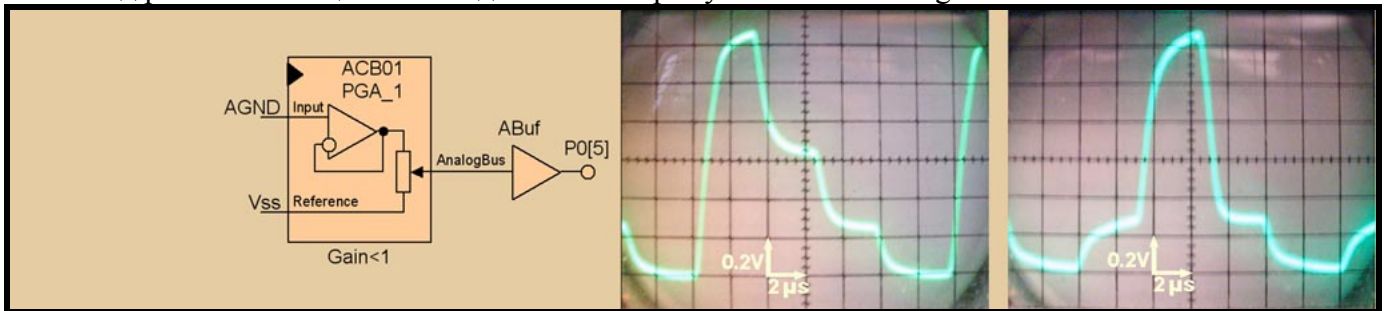
Согласно Техописанию (**Final Data Sheet No 38-12012 Rev. *G**) эта микросхема имеет следующие характеристики (при $U_{cc}=5V$ и параметрах $A_Buf_Power=High$ и $Op-Amp\ Bias=High$):

	Операц. усилители	Вых. буферы
Скорость нарастания (нагрузка 10 пФ, единичное усиление)	8.5 В/мкс	0.9 В/мкс
Скорость спада (нагрузка 10 пФ, единичное усиление)	6 В/мкс	0.9 В/мкс
Частота единичного усиления	8 МГц	
Полоса пропускания для малого сигнала (20мВpp, 3dB, 100пФ)		1 МГц

Как видно, характеристики операционных усилителей, входящих в состав блоков СТВ и SCB, гораздо лучше, чем у выходных буферов ABuf.

Метод 1: Переключение отвода резистивной цепочки у модуля PGA

1.1 Отвод резистивной цепочки подключен напрямую к шине AnalogBus

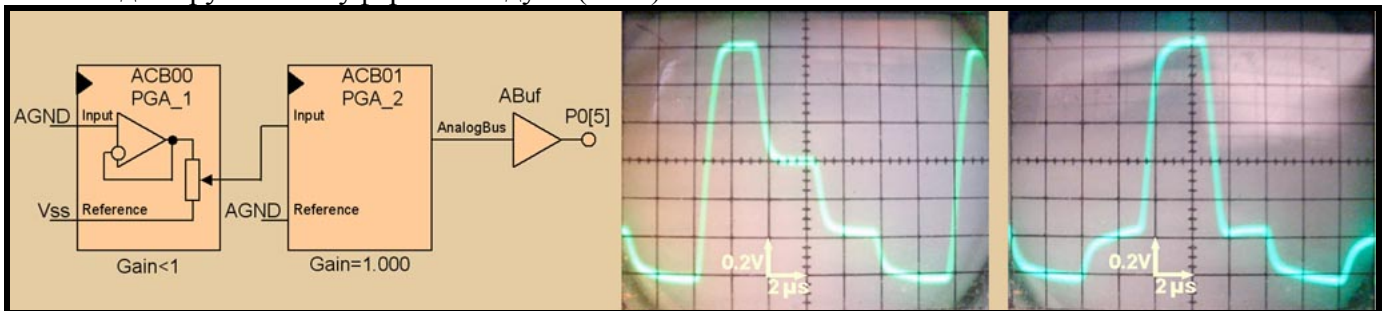


Ref Mux=(Vdd/2)+/(Vdd/2)

[Code 1](#) в Приложении

[Code 2](#) в Приложении

1.2 Отвод нагружен на буферный модуль (PGA)

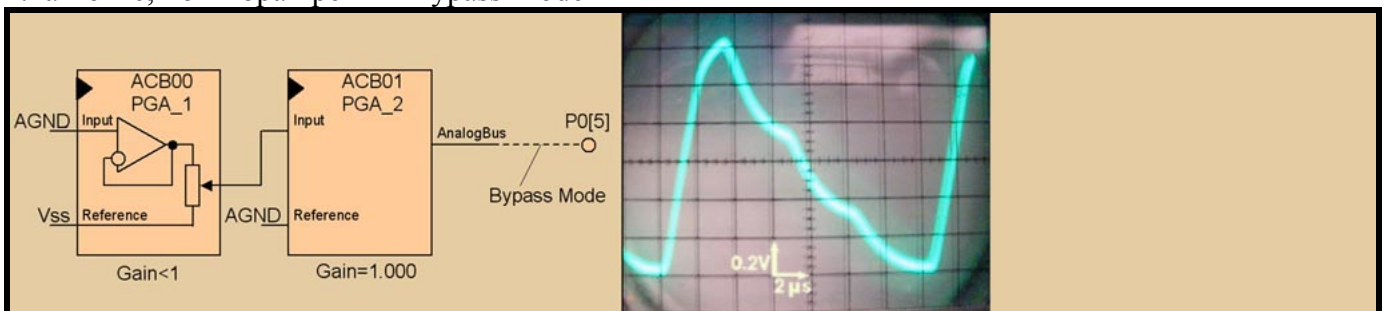


Ref Mux=(Vdd/2)+/(Vdd/2)

[Code 3](#) в Приложении

[Code 4](#) в Приложении

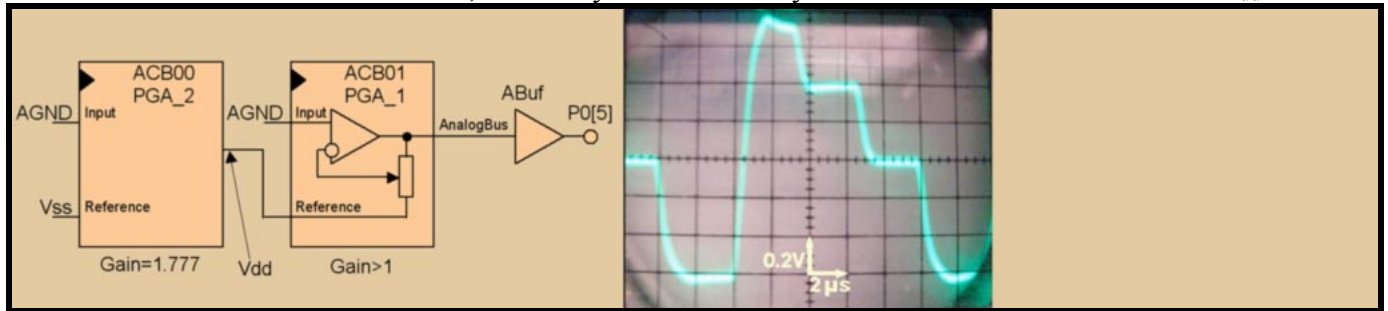
1.2a То же, но выбран режим Bypass Mode



Ref Mux=(Vdd/2)+/(Vdd/2)

[Code 5](#) в Приложении

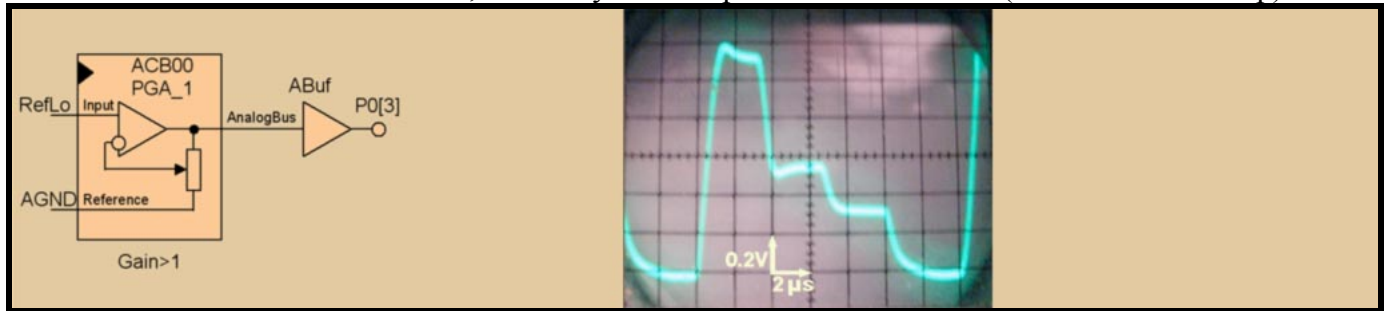
1.3 Отвод подключен в цепь ООС, используется доп. модуль PGA в качестве источника V_{dd}



$$\text{Ref Mux}=(V_{dd}/2)+/-(V_{dd}/2)$$

[Code 6](#) в Приложении

1.4 Отвод подключен в цепь ООС, используется опорный источник $\text{Ref}=(U_{dd}/2 \pm \text{BandGap})$

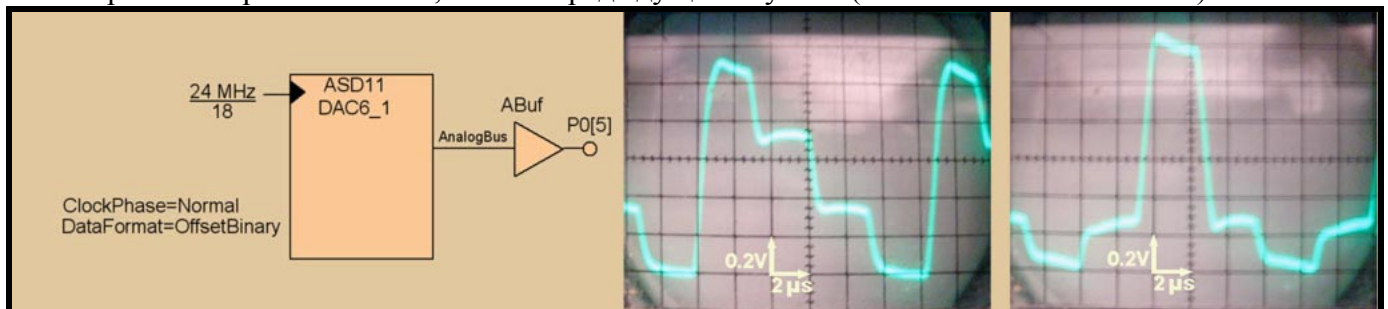


$$\text{Ref Mux}=(V_{dd}/2)+/-(\text{BandGap})$$

[Code 7](#) в Приложении

Метод 2: Прямая запись в регистры модуля DAC6

2.1 Скорость выбрана такой же, как и в предыдущих случаях (около 0.3М отсчетов/сек)

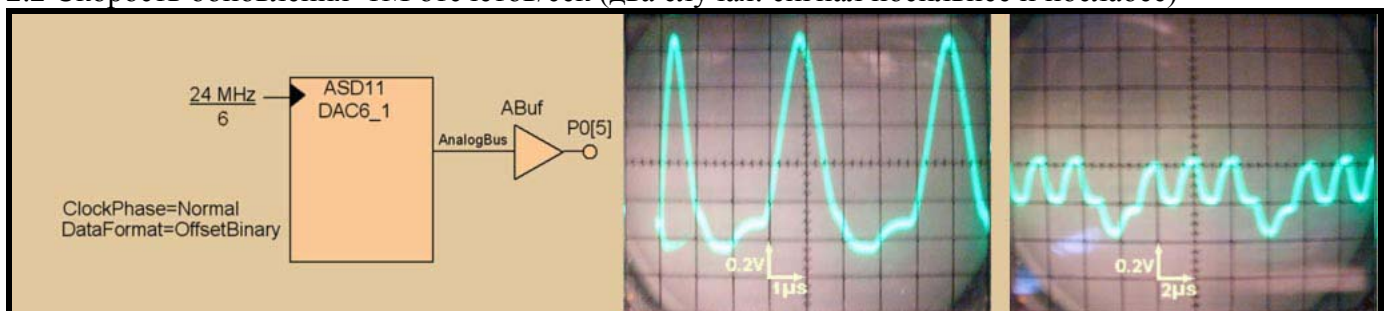


$$\text{Ref Mux}=(V_{dd}/2)+/-(V_{dd}/2)$$

[Code 8](#) в Приложении

[Code 9](#) в Приложении

2.2 Скорость обновления 1М отсчетов/сек (два случая: сигнал сильнее и послабее)



$$\text{Ref Mux}=(V_{dd}/2)+/-(V_{dd}/2)$$

[Code 10](#) in Appendix

[Code 11](#) in Appendix

Выводы

Предел скорости ограничен низким быстродействием выходных буферов. Нет ощутимой разницы в способе реализации, за исключением случая 1.1, когда отвод резистора нагружен на емкость шины Analog Bus. Режим Bypass Mode катастрофически затягивает фронты импульсов.

Приложение – Примеры кодов

Code 1	Code 3	Code 5
<pre> ;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 </pre>	<pre> ;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 </pre>	<pre> ;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 </pre>

Code 6	Code 7	Code 8
<pre> ;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 </pre>	<pre> ;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 </pre>	<pre> ;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 </pre>

Code 2	Code 4	Code 9
<pre> ;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 </pre>	<pre> ;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 </pre>	<pre> ;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 </pre>

Code 10	Code 11
<pre> ;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],0b00000110;=27 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 </pre>	<pre> ;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 </pre>