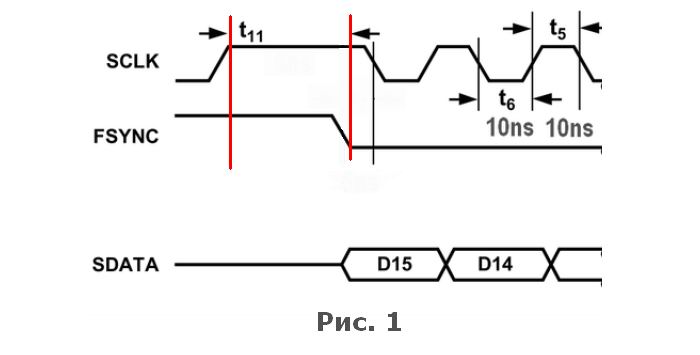
Инициализация микросхемы DDS AD9833

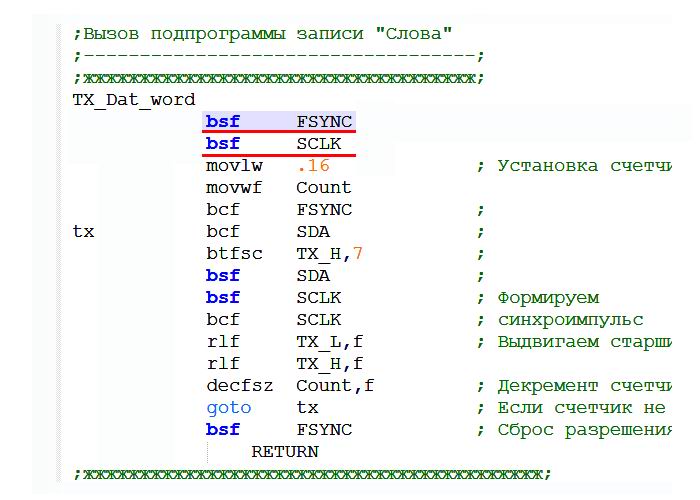
В статье рассматривается один из вариантов программной инициализации синтезатора частот AD9833 на языке Ассемблер для микроконтроллеров PIC16. Решил себе сделать цифровой НЧ генератор с выходным фильтром на переключаемых конденсаторах. И в качестве задающего генератора использовать данную микросхему DDS - устройства прямого цифрового синтеза аналоговых сигналов.

На данный момент я не буду вдаваться в теорию, что и как происходит внутри DDS синтезаторов, мне нужен задающий генератор синуса с малыми нелинейными искажениями. Если вы заинтересуетесь темой DDS, то в Сети куча информации. Сейчас пойдет разговор о том, как написать код для инициализации и ввода данных для AD9833 применительно к PIC контроллерам. Идея создания звукового НЧ генератора навеяна статьей Александра Корнева - «Цифровой перестраиваемый НЧ генератор синусоидального сигнала с низкими искажениями» размещенной в журнале РадиоЛоцман 01,02-2024. Что и когда все это воплотится в жизнь, я не знаю, но, по крайней мере, AD9833 уже работает совместно с PIC16F676.

Вообще не хотелось заморачиваться, думал найти все в Сети, но не тут-то было. Везде, где я побывал, везде технический перевод на микросхему и ардуино. Мне нужна была компактная схемка, недорогая в реализации, а не хрень с кучей проводов. Пришлось писать программу взаимодействия AD9833 и PIC16 самому. Но, как оказалось, ни один из кодов протокола передачи данных SPI, которыми я пользовался ранее, с AD9833 не работал. Сразу в процессе проб и ошибок возникло ощущение, что микросхема «не слышит, что к ней стучат». Но, все же заказал еще два модуля из разных магазинов. Когда посылка пришла, однозначно сделал вывод – надо анализировать тайминг протокола SPI в имеющихся у меня кодах. Оказалось, что для данной микросхемы важна последовательность начала передачи данных. На рисунке 1 между красными линиями.



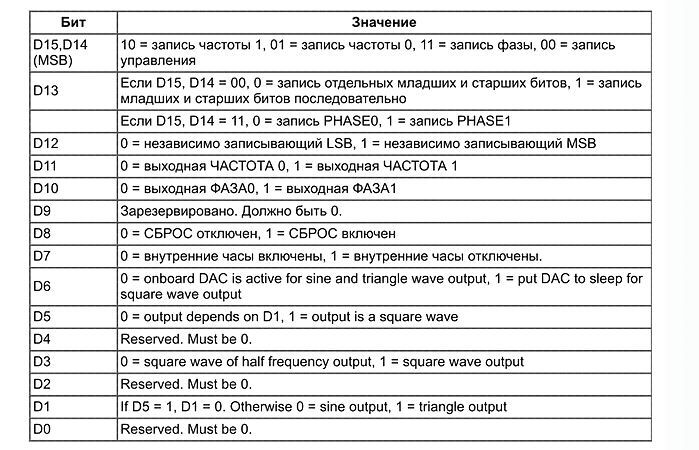
Спад FSYNC - сигнала разрешения обращения к микросхеме, всегда должен происходить при высоком уровне SCLK - сигнала тактирования. Ниже мой код (я не на что не претендую, явно, что кто то, где то такой уже написал, но как до его добраться) передачи по SPI шестнадцатибитного слова.



В моих старых кодах, заимствованных из Сети, отсутствовали две верхние подчеркнутые красными линиями строчки. Как раньше все работало, я не знаю. Да, та же MAX7219 – прекрасно работает и по сей день. Я здесь не стал использовать флаг «С», а решение на передачу нужного бита (0 или 1) принимается после анализа старшего бита 16и битного слова при последовательном смещении данных влево 16 раз. Такой код позволяет передавать данные практически любой разрядности. Например, записав в счетчик битов число 24 и добавив в код еще один регистр данных, например TX\_HH, получим передачу 24х битного или 3х байтового слова. Если выравнивать данные по левому краю, то можно передавать «любоебитное» слово.

Инициализация AD9833

Инициализация микросхемы обязательно начинается со сброса всех внутренних регистров DDS в нуль и выбора способа загрузки данных в 28-ми битовый регистр частоты. Загружать данные о частоте сигнала на выходе 10 – Vout AD9833 можно двумя способами, это зависит от состояния Бита 13 в регистре конфигурации.



Мне нужны прямоугольные импульсы и поэтому я буду описывать настройку конфигурации DDS применительно к этим условиям.

И так, первоначально передаем слово **b’0010 0001 0000 0000’ ; D16 и В15** – нули говорят микросхеме, что следующие 14 бит надо записать в регистр конфигурации. **D13** – “1” сообщает микросхеме, что данные о выводимой частоте мы будем передавать в 28-ми битный регистр частоты последовательно друг за другом за один раз словами по 16 бит, сначала младшее слово, потом старшее. Слова 2х байтовые. **D8 – “1”** – сброс регистров в нулевое состояние. Младшее слово у меня состоит из регистра Kf\_Lh и Kf\_Ll. Kf\_Lh – коэффициент частоты, младшее слово, старший байт. Kf\_Ll – коэффициент частоты, младшее слово, младший байт. Старшее слово состоит из регистров Kf\_Hh и Kf\_Hl. Kf\_Hh - коэффициент частоты, старшее слово, старший байт, Kf\_Hl – коэффициент частоты, старшее слово, младший байт. Пока на этом остановимся и ознакомимся с коэффициентом частоты, это я так обозвал буржуйское обозначение FREQREG. fMCLK – у меня Fопорная; а

fout – Fсигнала; Для нас так понятнее. Так вот этот коэффициент высчитывается по формуле :

1)

Так как собрался сделать НЧ генератор с фильтром на переключаемых конденсаторах, а для такого фильтра необходима тактовая частот в 100 раз превышающая значение звуковой частоты, то при необходимой нам, например, частоты в 20 000Гц, синтезатор должен выдавать прямоугольные импульсы частотой в 2МГц. Опорная частота в модуле с Али равна 25Мгц. По формуле 1 высчитываем Kf.

= 21 474 836,48;

Не забывайте, рассчитывать будет контроллер в двоичном исчислении, поэтому цифры за запятой удаляем и получаем 21 474 836 – или 0х0147 AE14, или 1 0100 0111 1010 1110 0001 0100.

Вот это число надо разделить между старшим и младшим словом. Так как регистр частоты в микросхеме 28ми битный, а передавать будем последовательно двумя 16ти битными словами, то делить Kf будем следующим образом: 14 младших бит, выделенных синим, перезапишем в регистры младшего слова и получим вот такой вид данных в этих регистрах - 0010 1110 0001 0100. 15 бит и 14 бит – информационные. Если они имеют вид 01, то запись будет в регистр частоты нулевой - FREQ0 REG.

Ставим «1» в 14 бит и получаем значение младшего слова - 0110 1110 0001 0100. – 0х6E14, его мы будем передавать первым. Запишем: 0х6E в Kf\_Lh, а 0х14 в Kf\_Ll . Теперь перезапишем оставшиеся красные, старшие биты Kf в старшее слово, значение которого будет иметь следующий вид: 000001 0100 0111 10, для наглядности сгруппируем число по тетрадам – 0000 0101 0001 1110; Старшее слово нам также надо записать в FREQ0 REG. Поэтому в 14 бит также заносим единицу и получаем 0100 0101 0001 1110 – 0х451E. Запишем: 0х45 в Kf\_Hh, а 0х1E в Kf\_Hl. «Я, понятно объясняю?» ☺

Еще одно уточнение, если 14 и 15 биты при передаче информации в микросхему равны нулю - 00, то значение остальных бит 13… 0, записываются в регистр конфигурации. А если эти биты не имеют нулевого значения (01,10, 11), то данные младших бит заносятся в регистры данных.

Теперь посмотрим на код. Мы с вами уже определились, что будем передавать первоначально –

- **b’0010 0001 0000 0000’.** Если уже забыли, что означают единички – вернитесь выше. Переведем в шестнадцатеричное значение – 0х2100. Для передачи данных в DDS в программе используется пара регистров - TX\_L и TX\_H. Теперь код:

**movlw 0X21 ; Бит13 "1" - запись младших и старших битов последовательно**

**movwf TX\_H ; Бит8 "1" - СБРОС внутренних регистров DDS не затрагивающий**

**movlw 0X00 ; регистры конфигурации, частоты и фазы.**

**movwf TX\_L ;**

**Call TX\_Dat\_word ;**

Теперь нам надо передать данные в 28 битный регистр частоты двумя 16-ти битными словами, младшим словом вперед.

**; Передача значения частоты - младшее слово**

**movfw Kf\_Lh ; =**0х6E

**movwf TX\_H ;**

**movfw Kf\_Ll ; =**0х14

**movwf TX\_L ;**

**Call TX\_Dat\_word ;**

**; Передача значения частоты - старшее слово**

**movfw Kf\_Hh ; =**0х45

**movwf TX\_H ;**

**movfw Kf\_Hl ; =**0х1E

**movwf TX\_L ;**

**Call TX\_Dat\_word ;**

С помощью данного кода мы передали данные, в соответствии с которыми DDS синтезирует частоту 2МГц. Так как мне нужна просто импульсная последовательность, без изменения фазы выходного сигнала, то нам необходимо в 16-ти битный регистр фазы записать нули. Данные будут иметь следующий вид: b’**11**00 0000 0000 0000’ , где **11 –** говорят о том, что остальные 14 бит - 13… 0 надо записать в регистр фазы - PHASEx.

; Передача значения фазы

movlw 0xC0 ; = **11**00 0000

movwf TX\_H ;

movlw 0x00 ; = 0000 0000

movwf TX\_L ;

Call TX\_Dat\_word ;

Теперь нам надо вернуться к регистру конфигурации и загрузить в него b’0000 0000 0110 1000’ – 0x00 68

Биты D15 и D14 – 00 – запись в регистр конфигурации, D8 – 0 – запрет сброса регистров. D6 – 1 - ЦАП переведен в спящий режим для вывода прямоугольной волны. D5 - 1 - выходное напряжение представляет собой прямоугольную волну. D3 – 1 - выходная частота прямоугольных импульсов не делится на 2.

**; Передача команды на зывершение**

**movlw 0x00 ;**

**movwf TX\_H ;**

**movlw 0x68 ;**

**movwf TX\_L ;**

**Call TX\_Dat\_word ;**

**Goto $ ;**

Всё. После выполнения данного кода на выходе DDS мы получаем импульсы частотой 2МГц. Форма импульсов на такой частоте конечно уже не столь прямоугольная. В схеме генератора НЧ из журнала предусмотрено деление этой частоты на 100 с помощью делителя 74HC390. Я пока до него не дошел, но думаю, что все будет работать.

Думаю, что основное понятие вы уже имеете, а получить другие виды сигнала в соответствии с таблицей сможете уже сами. В Сети очень много информации на эту тему, так что сложностей быть не должно.

На этом все. Успехов. К.В.Ю.