

Как с помощью независимой от ядра периферии микроконтроллеров PIC синтезировать новую функцию

Л. Ди Джасио¹

УДК 621.3.049.774 | ВАК 05.27.01

Разработчикам встраиваемых систем доступны разнообразные продукты и решения, позволяющие реализовать все многообразие функций. Однако зачастую выбор оптимального решения в процессе работы над проектом может быть обусловлен необходимостью перехода на другую модель микроконтроллера. Микроконтроллеры PIC от Microchip отличаются такими усовершенствованными функциями, как переназначение выводов микросхемы и независимая от ядра периферия, которые предоставляют разработчику гибкие возможности модификации проектного решения. На примере создания ШИМ-функции рассмотрим, как без дополнительных внешних компонентов и необходимости изменения топологии печатной платы можно быстро синтезировать новую автономную функцию периферии микроконтроллера.

ВВЕДЕНИЕ

Программисты встраиваемых систем и программисты компьютерных систем общего назначения различаются тем, что первые имеют дело с невероятным разнообразием аппаратных средств. Они, как летчики-испытатели, тогда как обычные пилоты управляют одним и тем же самолетом зачастую по одному и тому же маршруту. Программисты встраиваемых систем наслаждаются разнообразием машин и конфигураций, которые приходится часто менять в целях оптимизации стоимости, скорости и удобства использования под влиянием рыночного спроса и/или желаний заказчиков. Эта работа не бывает скучной, однако требует от программиста значительных усилий и затрат времени на поиск оптимального решения. С одной стороны, разработчики встраиваемых систем стремятся быстрее опробовать новейшие и передовые продукты. С другой стороны, любые инновации сопровождаются обучением, поэтому программисты и разработчики хотят, чтобы все пользовались одними и теми же компилятором, библиотекой и набором инструментов, с которыми они уже знакомы.

На рынке представлен широчайший выбор микроконтроллеров, разработчикам встраиваемых систем доступно любое количество контактов, разрядов, любая архитектура и практически любая комбинация флеш-памяти и ОЗУ. Конкурентная борьба среди основных поставщиков микроконтроллеров в последние годы была сфокусирована на том, чтобы добавить функций чипу и еще больше расширить матрицу выбора. Сегодня для создания встраиваемых

приложений важно, чтобы микроконтроллер был оснащен богатым набором периферии (функций).

Зачастую разработчики приступают к проекту с заданной моделью микроконтроллера, но вскоре понимают, что должны заменить ее другой прямо в середине проекта. И это случается как с ленивыми инженерами, которые вовремя не провели анализ, так и с теми, кто старательно готовился. Чаще это происходит из-за того, что в проекте нужно учитывать множество факторов, как в игре в шахматы. Разработчику может быть трудно прогнозировать, как периферийные функции будут взаимодействовать в приложении или как нужно модифицировать аппаратное решение, когда требования к проекту меняются (как всегда и случается) и впереди маячит срок сдачи проекта.

Смена модели микроконтроллера может оказаться наименьшим из зол, когда близкий аналог есть в том же семействе того же производителя. Сложнее, когда нужная опция недоступна, и вынужденный переход на другое семейство микроконтроллеров часто производит эффект домино, когда меняется цоколевка выводов микросхемы, а возможно, и другие периферийные функции.

РЕШЕНИЕ – НЕЗАВИСИМАЯ ОТ ЯДРА ПЕРИФЕРИЯ

Оптимальный способ решения этих проблем – заранее планировать возможность их появления. В новейших семействах микроконтроллеров предусмотрены умные способы расположить в ином порядке выводы для каждой периферийной функции. Причем не изменить функции отдельного вывода, а переназначить выводы всех аналоговых и цифровых функций. Например, предусмотренная в 8-, 16- и 32-разрядных микроконтроллерах Microchip

¹ Компания Microchip Technology, менеджер по развитию бизнеса в регионе EMEA.

специальная функция Peripheral Pin Select (PPS) обеспечивает возможность использования каждого вывода микросхемы в качестве входа для любого количества устройств периферии (в некоторых моделях до 64 периферийных функций), а любым выходом периферии может быть назначено до 16 выводов микросхемы (в некоторых моделях до 32). Аналоговые входы АЦП можно таким же образом распределить среди большей части выводов микросхемы (например, до 35 из 40 выводов чипа).

Это хорошая страховка от возможной несовместимости выводов при переходе на другие модели, но все же качественно новое решение, реализованное в микроконтроллерах Microchip, – независимая от ядра периферия (Core Independent Peripherals – CIP). Причем основная цель создания CIP – снижение рабочей нагрузки центрального процессора микроконтроллера за счет автоматизации как можно большего количества низкоуровневых и координационных операций, которые обычно требуются периферийным функциям. Однако благодаря доступности дополнительных взаимосвязей и гибким возможностям, CIP – прекрасный способ решения проблемы перехода на другую модель микроконтроллера в середине работы над проектом.

ПРИМЕР СОЗДАНИЯ НОВОЙ ШИМ-ФУНКЦИИ «ИЗ НИЧЕГО»

Рассмотрим случай из практики, когда нужно было спроектировать усовершенствованную версию автомобильного датчика с несколькими таймерами, парой 10-разрядных ШИМ-модулей и аналоговым компаратором. Все эти функции доступны в недорогом 28-выводном микроконтроллере PIC16F1885. Разработчики быстро поняли, что в текущем проектном решении достигнуто ограничение по разрешению, для преодоления которого потребовался 12-разрядный (или более высокой разрядности) ШИМ-регистр. К сожалению, этой опции не было предусмотрено в данной модели микроконтроллера. В аналогичных 8-разрядных микроконтроллерах PIC с совместимым форм-фактором (и цоколевкой выводов) предлагалась слишком маленькая

память программ либо намного более богатый набор периферии по более высокой цене. Все, что было нужно разработчикам, – дополнительный ШИМ-регистр с 12-разрядным или более высоким разрешением.

К счастью, ШИМ-функцию легко синтезировать на чипе при наличии некоторого количества конфигурируемых блоков и связующих логических схем. Любая ШИМ-функция может быть создана путем комбинации двух элементов: периодического импульсного сигнала и граничного счетчика. На рис. 1 показано, как можно комбинировать эти два элемента, чтобы получить прямоугольный сигнал с нужным периодом и коэффициентом заполнения (рабочим циклом).

НАСТРОЙКА ВРЕМЕННОЙ ДИАГРАММЫ

Временную диаграмму можно реализовать с помощью любого стандартного таймера, поскольку он формирует периодический импульсный сигнал нужной частоты (в нашем случае порядка 500 Гц). Для данного приложения в качестве таймера подходит 8-разрядный базовый таймер/счетчик, который может работать совместно с другими функциями (ШИМ и др.). В данном случае использовался модуль Timer2 – 8-разрядный базовый таймер с регистром совпадения. Конфигурирование модуля Timer2 в качестве базового таймера требует инициализации всего пары регистров, как показано на рис. 2.

Традиционный путь разработки такой подпрограммы инициализации потребовал бы детального изучения документации к выбранному микроконтроллеру PIC и тщательной идентификации битовой комбинации для каждого регистра. Процесс может занять длительное время и подвержен ошибкам даже в случае простой периферии. Сегодня доступен более оптимальный путь – использовать MPLAB Code Configurator, быстрый инструмент разработки (плагин), интегрированный в среду MPLAB X IDE, который позволяет генерировать код (например, подпрограмму инициализации, подобную приведенной на рис. 2) с помощью всего нескольких кликов мыши. По сути, этот инструмент часто используется для исследования функций устройств

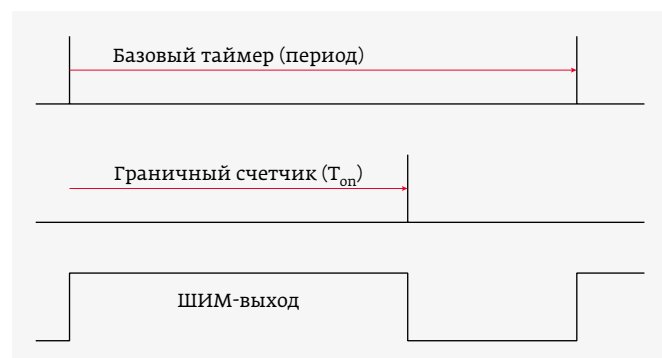


Рис. 1. ШИМ-функция как сумма сигналов базового таймера и граничного счетчика

```
void TMR2_Initialize(void)
{
    // T2CKPS 1:64; T2OUTPS 1:1; TMR2ON off
    T2CON = 0x60;

    // T2CS FOC5/4
    T2CLKCON = 0x01;

    // T2 period
    T2PR = 0xFF;

    // Clearing Interrupt flag
    PIR4bits.TMR2IF = 0;

    // Start TMR2
    T2CONbits.TMR2ON = 1
}
```

Рис. 2. Код инициализации модуля Timer2

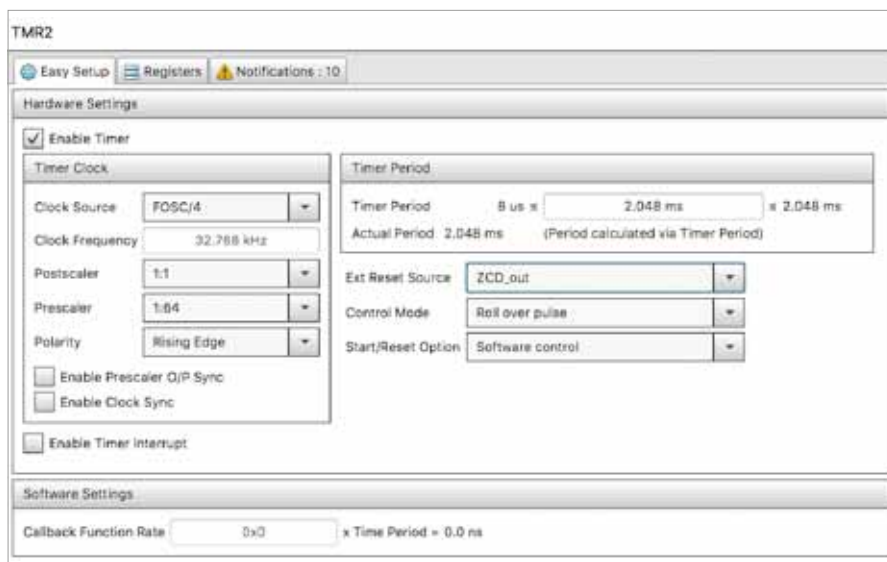


Рис. 3. Диалоговое окно MPLAB Code Configurator для конфигурирования модуля Timer2

и экспериментов со множеством опций, реализованных в независимой от ядра периферии.

На рис. 3 показаны доступные разработчику блоки и поля ввода диалогового окна MPLAB Code Configurator, с помощью которых можно выбрать все опции модуля Timer2.

КОНФИГУРИРОВАНИЕ ГРАНИЧНОГО СЧЕТЧИКА

Граничный счетчик – немного более сложная функция. После запуска таймер должен досчитать до некоторого граничного значения, сгенерировать импульс и остановиться. Он выполнит сброс и вновь начнет отсчет, только если на вход поступит следующий сигнал запуска. Эту функцию часто называют мультивибратором с повторным запуском, в литературе по микроконтроллерам PIC функция известна как аппаратный граничный таймер (Hardware Limit Timer – HLT).

К сожалению, для реализации 12-разрядной ШИМ-функции (или более высокой разрядности) типовой HLT-таймер, доступный на микроконтроллере PIC, не подойдет. В современных микроконтроллерах PIC

HLT-таймеры обеспечивают только 8-разрядные счетчики. Однако другое, менее известное, периферийное устройство может обеспечить нужное разрешение.

Это таймер измерения сигналов (Signal Measurement Timer – SMT), блок-схема которого показана на рис. 4.

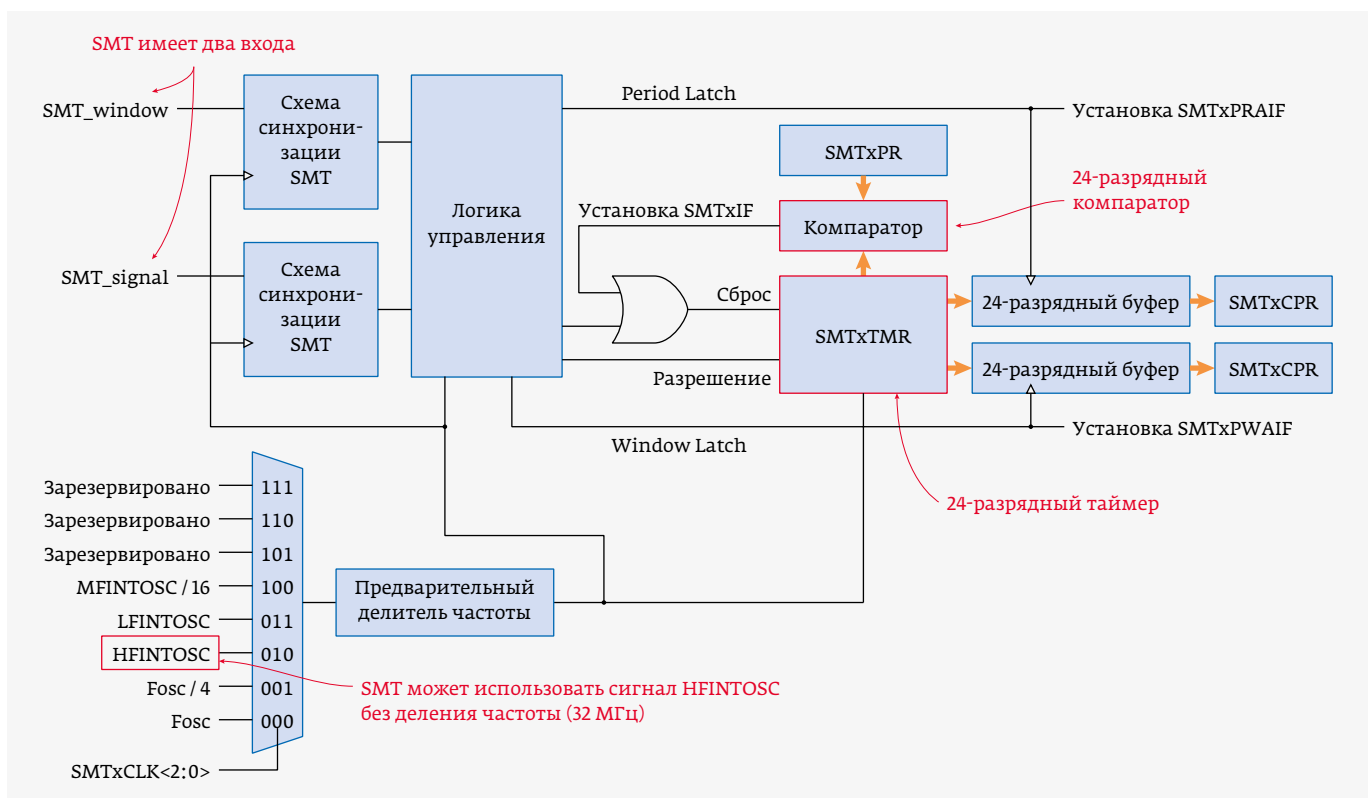


Рис. 4. Блок-схема таймера измерения сигналов (SMT)

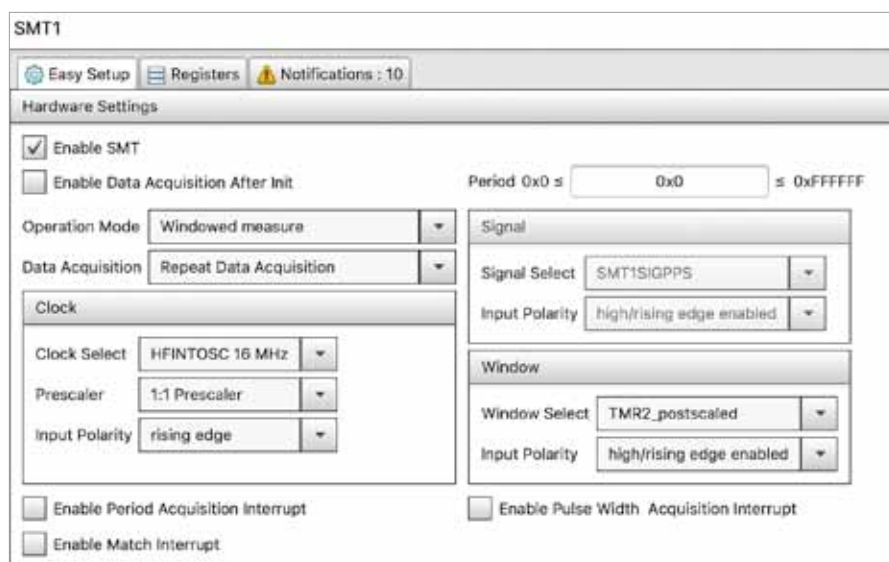


Рис. 5. Диалоговое окно МСС для конфигурирования модуля SMT1

Если HLT – таймер с одним дополнительным входом запуска / сброса, то SMT – таймер с двумя такими входами. Есть ряд интересных способов комбинирования входов для того, чтобы обеспечить измерения таких параметров сигнала, как время, частота, коэффициент заполнения – отсюда и название функции.

В документации к микроконтроллеру представлен список из 11 режимов измерения, что может напугать неопытного пользователя и сделать SMT самым непонятным и недоиспользованным устройством среди CIP-периферии. Один из этих режимов – режим оконных измерений (Window Measurement Mode) – фактически превращает SMT в HLT. По сути, SMT – это очень крупный HLT-таймер, поскольку содержит 24-разрядный счетчик и регистр периода соответствующего размера. Конфигурирование SMT-таймера (SMT1) аналогичным образом можно значительно упростить благодаря использованию инструмента МСС, как показано на рис. 5.

С помощью нескольких кликов мыши можно автоматически сгенерировать функцию инициализации (рис. 6), аналогичную функции инициализации TMR2 (см. рис. 2).

КОНФИГУРИРОВАНИЕ S / R-ТРИГГЕРА

Для комбинации функций базового таймера и граничного счетчика можно использовать триггер установки / сброса (S / R-триггер). Эта функция, доступная на многих микроконтроллерах PIC, за последние пять лет была интегрирована в качестве одной из восьми функций в конфигурируемые логические ячейки (Configurable Logic Cell – CLC) – один из основных элементов CIP-периферии.

На рис. 7 показано, как эти два сигнала можно подать на входы S / R-триггера, чтобы получить ШИМ-выход

(на рисунке представлено диалоговое окно MPLAB Code Configurator для конфигурирования CLC-модулей).

Используется быстрый инструмент разработки (МСС) или конфигурируется периферия вручную, результат один и тот же – компактная функция инициализации, содержащая всего несколько операций присваивания для соответствующих регистров периферии (рис. 8).

Поскольку микроконтроллеры PIC16F18xx в нашем приложении оснащены двумя SMT-модулями, можно было синтезировать не один, а два новых ШИМ-модуля с требуемым удвоенным разрешением.

Дополнительные несколько строк кода завершат конфигурирование новой ШИМ-функции и позволят

контролировать ее коэффициент заполнения (рабочий цикл). На рис. 9 показано, как можно реализовать инициализацию периферии путем простого связывания функций инициализации трех используемых устройств периферии. Запуск и остановка нового ШИМ-модуля теперь управляется граничным счетчиком (SMT1), а коэффициент заполнения контролируется 24-разрядным регистром периода SMT. В этой процедуре перед изменением значения коэффициента заполнения был добавлен простой цикл синхронизации, чтобы избежать выбросов на выходе.

ЭФФЕКТИВНОЕ РАЗРЕШЕНИЕ ШИМ-МОДУЛЯ

Среди неопытных пользователей распространено ложное представление, что разрешение ШИМ-регистра определяется числом разрядов таймера, управляющего длительностью импульса в состоянии «включено» (коэффициентом заполнения). Согласно этому определению нам нужно было бы создать 24-разрядную ШИМ-функцию, но это не так.

```
void SMT1_Initialize(void)
{
    // WPOL high/rising edge enabled; SMT1EN enabled;
    // SMT1PS 1:1 Prescaler
    SMT1CON0 = 0x80;

    // SMT1MODE Windowed measure; SMT1GO disabled
    SMT1CON1 = 0x44;

    // SMT1CSEL HFINTOSC
    SMT1CLK = 0x02;

    // SMT1WSEL TMR2_postscaled;
    SMT1WIN = 0x05;
}
```

Рис. 6. Код инициализации модуля SMT1

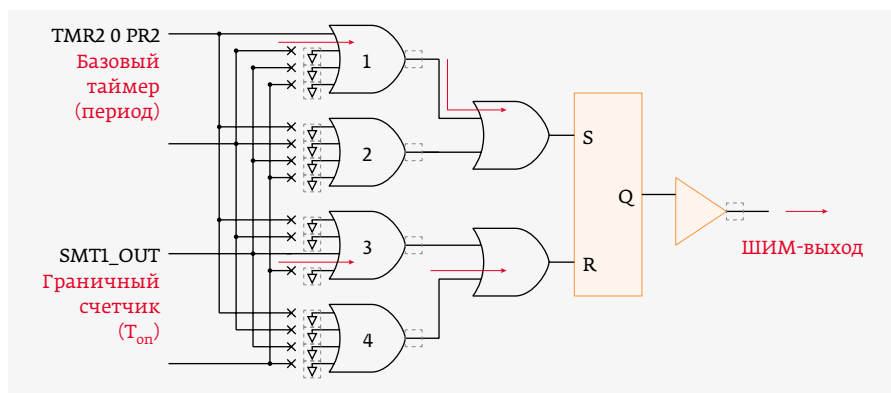


Рис. 7. Конфигурирование S/R-триггера с помощью MPLAB Code Configurator

Период и коэффициент заполнения ШИМ-функции – это не полностью независимые величины: как можно было увидеть выше, они контролируются таймерами/счетчиками и связаны с тактовым сигналом периферии микроконтроллера.

Для любого заданного тактового сигнала периферии счетчик, управляющий временем включения, способен досчитать только до значения *MaxCount*, прежде чем повторится очередной период временной диаграммы. В этой точке ШИМ-регистр достигнет 100%-ного коэффициента заполнения и дополнительные разряды в регистре не будут использоваться. Эффективное разрешение созданной в примере ШИМ-функции определяется по формуле:

$$MaxCount = T_{period} \cdot F_{clock}$$

В нашем случае для $T_{period} = 2,048$ мс и тактовой частоты периферии, равной 32 МГц, (SMT-триггер можно синхронизировать от максимальной частоты системного генератора, не ограниченной тактовой частотой команд ЦП, деленной на 4, как для основной части

периферии микроконтроллера PIC) $MaxCount = 65,356$.

Логарифм по основанию 2 от величины *MaxCount* равен эффективно разрешению ШИМ-функции в битах, которая в данном случае равна 16 битам.

Результат подтверждает, что новый ШИМ-модуль действительно отличается более высоким разрешением, чем стандартные 10-разрядные ШИМ-модули микроконтроллеров PIC, и наша цель даже превышена.

```
void PWM_Initialize ()
{
    TMR2_Initialize();
    SMT1_Initialize();
    CLC1_Initialize();
}

void PWM_Start(){
    SMT1CON1bits.SMT1GO = 1;
}

void PWM_Stop (){
    SMT1CON1bits.SMT1GO = 0;
}

void PWM_DutyCycleSet(uint32_t dc){
    while (CLC1_OutputStatusGet()==1);
    SMT1_SetPeriod(dc);
}
```

Рис. 9. Конфигурирование нового ШИМ-модуля и функций управления

ЗАКЛЮЧЕНИЕ

Отметим, что речь идет о создании новой эффективной 16-разрядной ШИМ-функции без использования дополнительных внешних компонентов. Мы просто сконфигурировали, соединили и повторно использовали небольшое количество функций, доступных на чипе. Более того, можно заменить созданную функцию в существующем решении без модификации топологии печатной платы, поскольку функция Peripheral Pin Select (PPS) позволяет реализовать новую функцию на любом выводе чипа. К тому же созданный модуль функционирует как полностью автономная функция (свойство CIP), поскольку для поддержки непрерывной ее работы не требуется ресурсов ЦП, за исключением начальной установки и периодической регулировки коэффициента заполнения (рабочего цикла).

ЛИТЕРАТУРА

1. PIC16F18855 datasheet. – <http://www.microchip.com/wwwproducts/en/PIC16F18855>
2. MPLAB Code Configurator. – <http://microchip.com/mcc>
3. CIPs: www.microchip.com/cip

```
void CLC1_Initialize(void)
{
    // LC1D1S TMR2=T2PR
    CLC1SEL0 = 0x0c;

    // LC1D4S SMT1_OUT
    CLC1SEL3 = 0x11;

    // LC1G1D1T enabled
    CLC1GLS0 = 0x02;

    // LC1G4D4T enabled
    CLC1GLS3 = 0x80;

    // LC1EN enabled; MODE SR latch
    CLC1CON = 0x83;
}
```

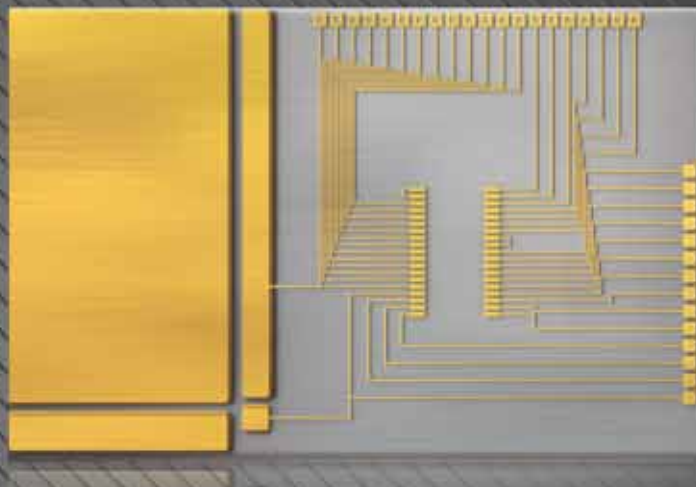
Рис. 8. Конфигурирование модуля CLC1 в качестве S/R-триггера

ИЗГОТОВЛЕНИЕ КЕРАМИЧЕСКИХ ПЛАТ И ПОДЛОЖЕК

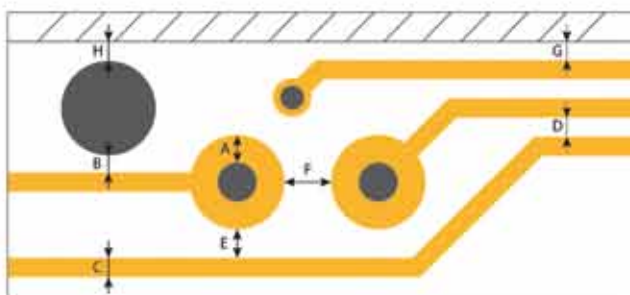
ТЕСТПРИБОР

ИСПОЛЬЗУЕМЫЕ МАТЕРИАЛЫ

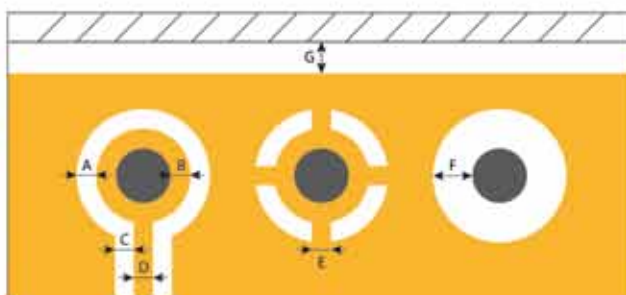
Al₂O₃ 96%
Al₂O₃ 99,6%
AlN



ТОПОЛОГИЧЕСКИЕ НОРМЫ СИГНАЛЬНЫХ СЛОЕВ



ТОПОЛОГИЧЕСКИЕ НОРМЫ СИЛОВЫХ СЛОЕВ



Наименование параметра	Керамические подложки с металлизацией на основе толстополеночной технологии	Керамические подложки с металлизацией на основе тонкополеночной технологии
Материал проводников и металлизации	W/Ni-Au или Mo/Ni-Au (Ni 5 мкм max, Au 0,5 мкм max)	TiW/Au; TaN/TiW/Au; TiW/Ni/Au; TaN/TiW/Ni/Au; TaN/NiW/Au/Cu/Ni/Au
Поверхностное сопротивление проводников	10,0 мОм/□	—
Сопротивление переходных отверстий размером (Ø0,2×0,25) мм	6,0 мОм	—
A	0,20	0,20/020
B	0,60	0,25/025
C	0,20	0,15/0,20
D	0,20/0,25	0,15/0,20
E	0,20/0,30	0,15/0,38
F	0,25/0,65	0,15/0,30
G	0,60/0,60	0,25/0,50
H	0,60/-	0,15/-

ИМС КАТЕГОРИИ КАЧЕСТВА «ВП» НИЗКОВОЛЬТНЫХ БЫСТРОДЕЙСТВУЮЩИХ ПРИЕМОПЕРЕДАТЧИКОВ ИНТЕРФЕЙСА LVDS 5560ИН13У И 5560ИН14У

Микросхемы 5560ИН13У и 5560ИН14У – низковольтные быстродействующие приемопередатчики интерфейса LVDS с напряжением питания 3,3 В для применения в аппаратуре специального назначения. Микросхемы изготавливаются в малогабаритных металлокерамических корпусах типа 5119.16-А и функционируют при температуре среды от -60 до $+125^{\circ}\text{C}$. Функциональными аналогами микросхем 5560ИН13У и 5560ИН14У являются соответственно микросхемы SN65LVDT179 и SN65LVDT180 компании Texas Instruments. Технические условия – АЕЯР431200.765-11 ТУ.

Таблица 1. Электрические параметры микросхем при приемке и поставке (при $U_{\text{cc}} = 3,0 + 3,6 \text{ В}$)

Условное обозначение микросхемы	Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
			не менее	не более	
1	2	3	4	5	6
5560ИН13У 5560ИН14У	Ток потребления, мА, активный режим, приемник не нагружен, нагрузка передатчика $R_L = 100 \text{ Ом}$	I_{CC1}	—	12	25 ± 10 , —60, 125
	передатчик разрешен, приемник выключен, нагрузка передатчика $R_L = 100 \text{ Ом}$	I_{CC2}	—	7,0	
	передатчик выключен, приемник разрешен, без нагрузки	I_{CC3}	—	3,0	
	микросхема выключена	I_{CC4}	—	1,0	
	Выходной ток низкого уровня в состоянии «Выключено» при $U_O = 0 \text{ мкА}$	I_{OZL}	—	—10	
	Выходной ток высокого уровня в состоянии «Выключено» при $U_O = U_{CC}$, мкА	I_{OZH}	—	10	
Электрические параметры передатчика					
5560ИН13У 5560ИН14У	Выходное напряжение низкого уровня при $I_{OL} = 8,0 \text{ мА}$, В	U_{OL}	—	0,4	25 ± 10 , —60, 125
	Выходное напряжение высокого уровня, В	U_{OH}	2,4	—	
	$I_{OH} = -8,0 \text{ мА}$		2,8	—	
	$I_{OH} = -4,0 \text{ мА}$	I_i	—2,0	—20	
	Входной ток по входам А или В, мкА		—1,2	—	
	$U_i = 0$	I_{ILR}	—	10	
$U_i = 2,4 \text{ В}$	—		10		
5560ИН14У	Входной ток низкого уровня по входам разрешения при $U_{iL} = 0,8 \text{ В}$, мкА	I_{ILR}	—	10	25 ± 10 , —60, 125
	Входной ток высокого уровня по входам разрешения при $U_{iH} = 5,0 \text{ В}$, мкА	I_{IHR}	—	10	
	Входной ток при выключенном питании по входам А или В при $U_{CC} = 0$, мкА	$I_{i(OFF)}$	—	±20	
5560ИН14У	Дифференциальный входной ток, мА $U_{iD} = 0,4 \text{ В}$, $U_{iC} = 0,2 \text{ В}$ $U_{iD} = 0,4 \text{ В}$, $U_{iC} = 2,2 \text{ В}$	I_{iD}	3,0	4,4	

Продолжение таблицы 1.

Условное обозначение микросхемы	Наименование параметра, единица измерения, режим измерения	Буквенное обозначение параметра	Норма параметра		Температура среды, °C
			не менее	не более	
Электрические параметры передатчика					
5560ИН13У 5560ИН14У	Выходное дифференциальное напряжение, мВ, R _L = 100 Ом	U _{OD}	±247	±454	25 ± 10, -60, 125
	Разность выходных дифференциальных напряжений, мВ, R _L = 100 Ом	ΔU _{OD}	-50	50	
	Выходное напряжение смещения относительно общего вывода, В, R _L = 49,9 Ом	U _{OC}	1,125	1,375	
	Разность выходных напряжений смещения относительно общего вывода, мВ, R _L = 49,9 Ом	ΔU _{OC}	-50	50	
	Размах выходного напряжения смещения относительно общего вывода при переключении, мВ, R _L = 49,9 Ом, C _L = 10 пФ	U _{OC PP}	-	150	
	Входной ток низкого уровня по входам D, DE при U _{IL} = 0,8 В, мкА	I _{ILD}	-	10	
	Входной ток высокого уровня по входам D, DE при U _{IH} = 5,0 В, мкА	I _{IHD}	-	20	
	Ток короткого замыкания, мА, U _{OY} или U _{OȲ} = 0	I _{OS}	-	-10	
	U _{OD} = 0		-	10	
	Выходной ток в выключенном состоянии, мкА, U _{DE} = 0, U _{OY} = U _{OȲ} = 0	I _{O(OFF)}	-	±1,0	
	U _{CC} < 1,5 В, U _{DE} = U _{CC} , U _{OY} = U _{OȲ} = 0		-	±1,0	

Таблица 2. Таблица истинности передатчика

Входы		Выходы	
D	DE	Y	\bar{Y}
H	H	H	L
L	H	L	H
Открыт	H	L	H
X	L	Z	Z

H – высокий уровень напряжения;
 L – низкий уровень напряжения;
 X – низкий или высокий уровень напряжения;
 Z – выход в состоянии «Выключено»

Таблица 3. Таблица истинности приемника

Входы		Выход
$U_{ID} = A - \bar{B}$	RE	R
$U_{ID} \geq +100 \text{ мВ}$	L	H
$-100 \text{ мВ} \leq U_{ID} \leq +100 \text{ мВ}$	L	*
$U_{ID} \leq -50 \text{ мВ}$	L	L
Открыт	L	H
X	H	Z

H – высокий уровень напряжения;
 L – низкий уровень напряжения;
 X – низкий или высокий уровень напряжения;
 Z – выход в состоянии «Выключено»
 * – неопределенное состояние

Микросхемы 5560ИН13У и 5560ИН14У должны быть стойкими к воздействию специальных факторов 7.И, 7.К по ГОСТ РВ 20.39.414.2 с характеристиками 7.И1 – 4Ус; 7.И6 – 4Ус; 7.И7 – 4Ус; 7.К1 – 2К; 7.К4 – 1К, 7.К11 (7.К12) – до уровня 60 МэВ см²/мг по катастрофическим отказам и тиристорному эффекту.

Планируемый срок начала освоения в серийном производстве микросхем 5560ИН13У и 5560ИН14У – 1 кв. 2020 г.



ОАО «ИНТЕГРАЛ» –
 управляющая компания
 холдинга «ИНТЕГРАЛ»
 Республика Беларусь
 E-mail: ATitov@integral.by

тел./факс: (+375-17) 398 72 03
 тел.: (+375-17) 298 97 43
www.integral.by