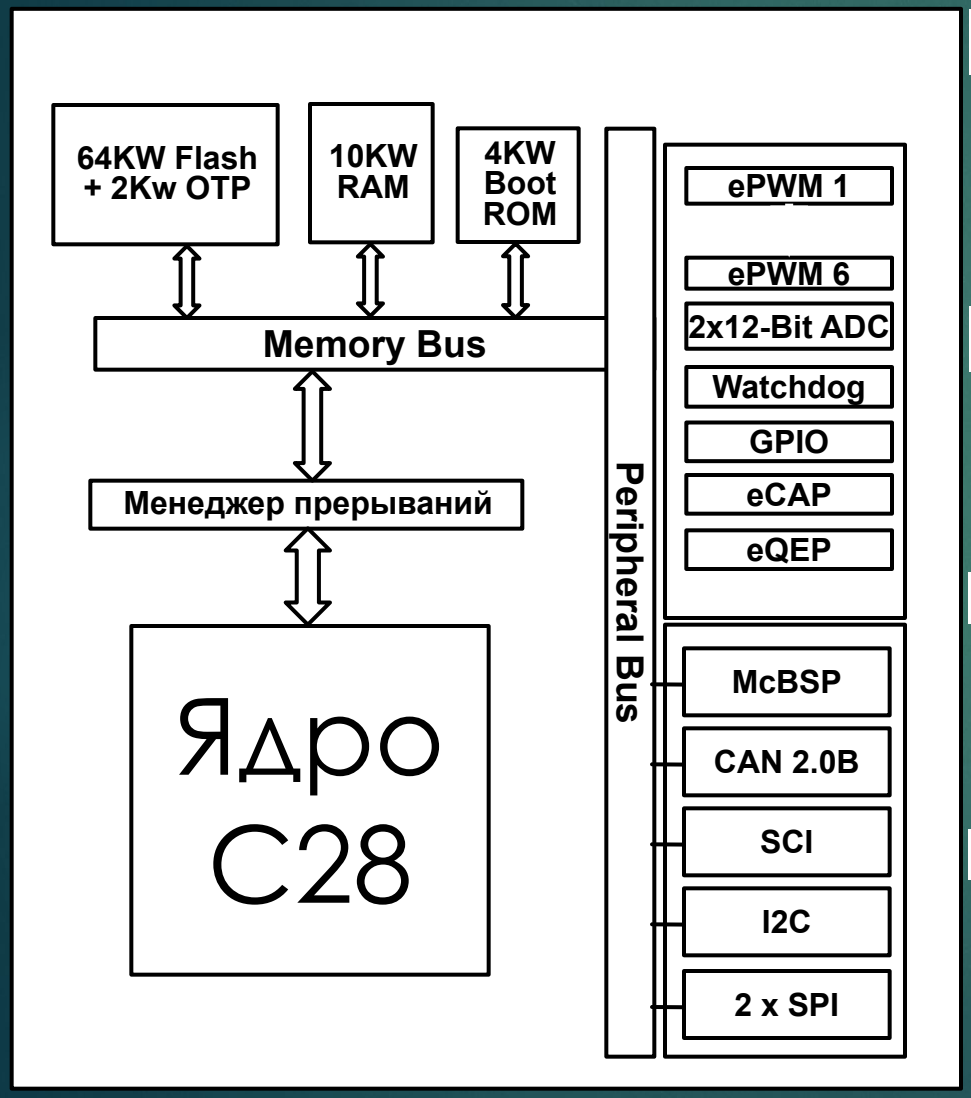
TMS320F28035
Состав периферийных устройств
Порты ввода/вывода
Менеджер прерываний

Состав периферийных устройств TMS320F28035



Подсистема встроенной памяти

Быстрое выполнение программы из ОЗУ или Флэш-памяти

- 60 MIPS по технологии акселерированной Флеш-памяти

Порты управления реального времени

12-разрядный АЦП

- 12.5 MSPS (млн.выб./с)
- Одновременная выборка двух сигналов

Коммуникационные порты

Несколько стандартных интерфейсов

- SPI, UART, CAN, I2C

Применения

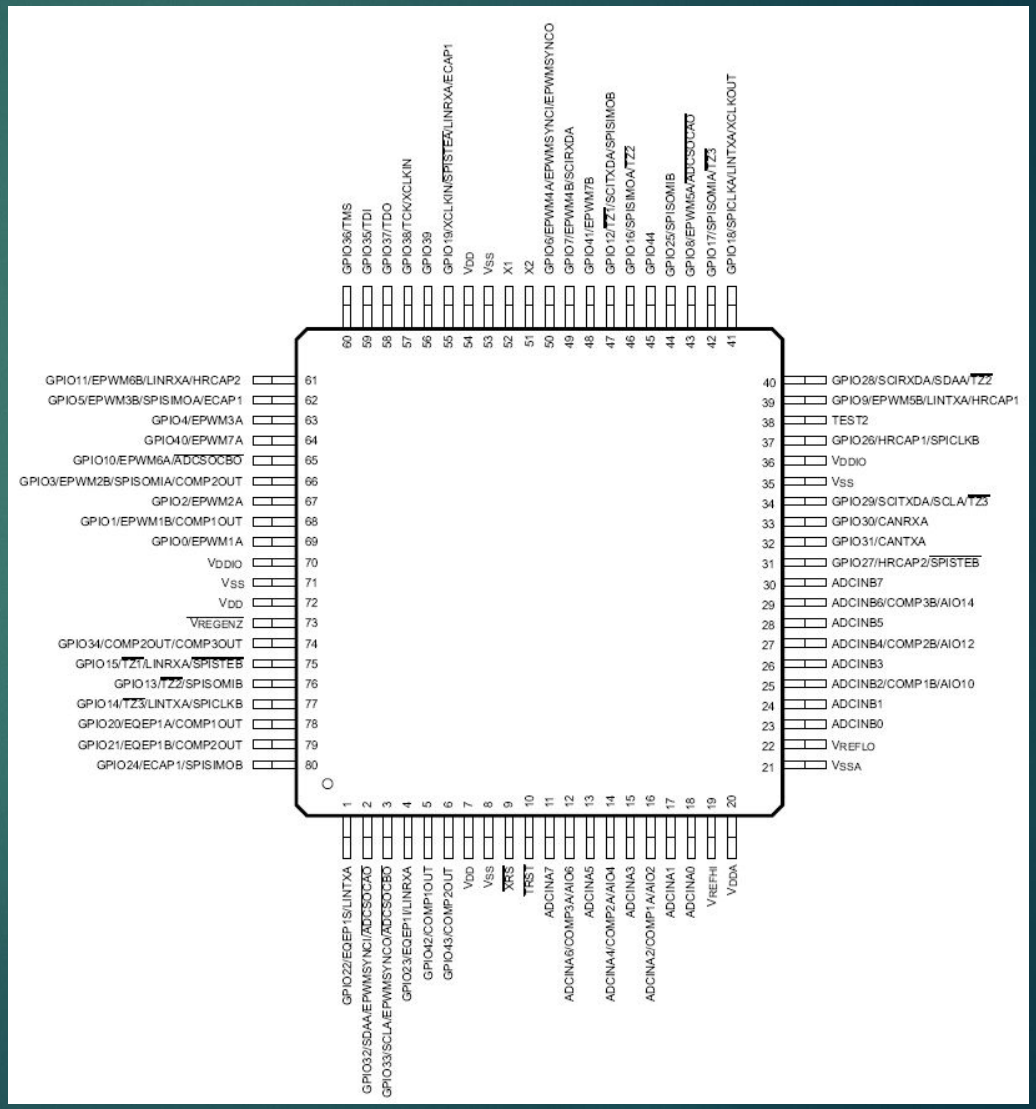
Управление двигателями, силовыми преобразователями, источниками питания, автоматизация технологических процессов

Состав периферийных устройств TMS320F28035

Микроконтроллер TI F28035 может иметь до 80 выводов в зависимости от выбранной модификации. Назначение выводов:

- питание
- тактирование
- подключение эмулятора
- ввод аналоговых сигналов
- вводы/выводы общего назначения

Вводы/выводы общего назначения в свою очередь могут управляться либо пользовательской программой, встроенным периферийным устройством.



Менеджер событий TMS320F28035 Состав модулей, выходы, входы

- 6 модулей **ePWM** (Enhanced Pulse-Width Modulator) для генерации ШИМ-сигнала
- 1 модуль **eCAP** (Enhanced Capture) для обработки датчиков Холла
- 1 модуль **eQEP** (Enhanced Quadrature Encoder Pulse) для обработки квадратурных энкодеров
- 1 модуль **ADC** (Analog-to-Digital Converter) для обработки аналоговых сигналов
- Модули связи: 2 x **SPI** (Serial Peripheral Interface), **I2C** (Inter-Integrated Circuit), **SCI** (Serial Communications Interface), **LIN** (Local Interconnect Network), **eCAN** (Controller Area Network)
- 1 сопроцессор **CLA** (Control Law Accelerator) для параллельных вычислений
- 3 процессорных **таймера** общего назначения (32-разряда)
- 45 **GPIO** (General Purpose Input-Output) – выводы общего назначения

TMS320F28035
Порты ввода/вывода

Регистр GPAMUX1

GPAMUX1 Register Bits	Default at Reset Primary I/O Function (GPAMUX1 bits = 00)	Peripheral Selection (GPAMUX1 bits = 01)	Peripheral Selection 2 (GPAMUX1 bits = 10)	Peripheral Selection 3 (GPAMUX1 bits = 11)
1-0	GPIO0	EPWM1A (O)	Reserved	Reserved
3-2	GPIO1	EPWM1B (O)	Reserved	COMP1OUT (O)
5-4	GPIO2	EPWM2A (O)	Reserved(1)	Reserved(1)
7-6	GPIO3	EPWM2B (O)	SPISOMIA (I/O)	COMP2OUT (O)
9-8	GPIO4	EPWM3A (O)	Reserved(1)	Reserved(1)
11-10	GPIO5	EPWM3B (O)	SPISIMOA (I/O)	ECAP1 (I/O)
13-12	GPIO6	EPWM4A (O)	EPWMSYNCI (I)	EPWMSYNCO (O)
15-14	GPIO7	EPWM4B (O)	SCIRXDA (I)	Reserved
17-16	GPIO8	EPWM5A (O)	Reserved	$\overline{\text{ADCSOCAO}}$ (O)
19-18	GPIO9	EPWM5B (O)	LINTXA (O)	Reserved
21-20	GPIO10	EPWM6A (O)	Reserved	$\overline{\text{ADCSOCBO}}$ (O)
23-22	GPIO11	EPWM6B (O)	LINRXA (I)	Reserved
25-24	GPIO12	$\overline{\text{TZ1}}$ (I)	SCITXDA (O)	SPISIMOB (I/O)
27-26	GPIO13(2)	$\overline{\text{TZ2}}$(I)	Reserved	SPISOMIB (I/O)
29-28	GPIO14(2)	$\overline{\text{TZ3}}$ (I)	LINTXA (O)	SPICLKB (I/O)
31-30	GPIO15(2)	$\overline{\text{TZ1}}$ (I)	LINRXA (I)	$\overline{\text{SPISTEB}}$ (I/O)

Регистр GPAMUX2

GPAMUX2 Register Bits	(GPAMUX2 bits = 00)	(GPAMUX2 bits = 01)	(GPAMUX2 bits = 10)	(GPAMUX2 bits = 11)
1-0	GPIO16	SPISIMOA (I/O)	Reserved	$\overline{\text{TZ2}}$ (I)
3-2	GPIO17	SPISOMIA (I/O)	Reserved	$\overline{\text{TZ3}}$ (I)
5-4	GPIO18	SPICLKA (I/O)	LINTXA (O)	XCLKOUT (O)
7-6	GPIO19/XCLKIN	$\overline{\text{SPISTEA}}$ (I/O)	LINRXA (I)	ECAP1 (I/O)
9-8	GPIO20	EQEP1A (I)	Reserved	COMP1OUT (O)
11-10	GPIO21	EQEP1B (I)	Reserved	COMP2OUT (O)
13-12	GPIO22	EQEP1S (I/O)	Reserved	LINTXA (O)
15-14	GPIO23	EQEP1I (I/O)	Reserved	LINRXA (I)
17-16	GPIO24	ECAP1 (I/O)	Reserved	SPISIMOB (I/O)
19-18	GPIO25 (2)	Reserved	Reserved	SPISOMIB (I/O)
21-20	GPIO26 (2)	Reserved	Reserved	SPICLKB (I/O)
23-22	GPIO27 (2)	Reserved	Reserved	$\overline{\text{SPISTEB}}$ (I/O)
25-24	GPIO28	SCIRXDA (I)	SDAA (I/OC)	$\overline{\text{TZ2}}$ (I)
27-26	GPIO29	SCITXDA (O)	SCLA (I/OC)	$\overline{\text{TZ3}}$ (I)
29-28	GPIO30	CANRXA (I)	Reserved	Reserved
31-30	GPIO31	CANTXA (O)	Reserved	Reserved

Регистр GPBMUX1

GPBMUX1 Register Bits	Default at Reset Primary I/O Function (GPBMUX1 bits = 00)	Peripheral Selection 1 (GPBMUX1 bits = 01)	Peripheral Selection 2 (GPBMUX1 bits = 10)	Peripheral Selection 3 (GPBMUX1 bits = 11)
1-0	GPIO32	SDAA (I/OC)	EPWMSYNCI (I)	$\overline{\text{ADCSOCAO}}$ (O)
3-2	GPIO33	SCLA (I/OC)	EPWMSYNCO (O)	$\overline{\text{ADCSOCBO}}$ (O)
5-4	GPIO34	COMP2OUT (O)	Reserved	COMP3OUT (O)
7-6	GPIO35 (TDI)	Reserved	Reserved	Reserved
9-8	GPIO36 (TMS)	Reserved	Reserved	Reserved
11-10	GPIO37 (TDO)	Reserved	Reserved	Reserved
13-12	GPIO38/XCLKIN (TCK)	Reserved	Reserved	Reserved
15-14	GPIO39	Reserved	Reserved	Reserved
17-16	GPIO40	EPWM7A (O)	Reserved	Reserved
19-18	GPIO41	EPWM7B (O)	Reserved	Reserved
21-20	GPIO42	Reserved	Reserved	COMP1OUT (O)
23-22	GPIO43	Reserved	Reserved	COMP2OUT (O)
25-24	GPIO44	Reserved	Reserved	Reserved
27-26	Reserved	Reserved	Reserved	Reserved
29-28	Reserved	Reserved	Reserved	Reserved
31-30	Reserved	Reserved	Reserved	Reserved

Регистры для настройки GPIO

Name	Address	Size (x16)	Register Description
GPADAT	0x6FC0	2	GPIO A Data Register (GPIO0-GPIO31)
GPASET	0x6FC2	2	GPIO A Set Register (GPIO0-GPIO31)
GPACLEAR	0x6FC4	2	GPIO A Clear Register (GPIO0-GPIO31)
GPATOGGLE	0x6FC6	2	GPIO A Toggle Register (GPIO0-GPIO31)
GPBDAT	0x6FC8	2	GPIO B Data Register (GPIO32-GPIO38)
GPBSET	0x6FCA	2	GPIO B Set Register (GPIO32-GPIO38)
GPBCLEAR	0x6FCC	2	GPIO B Clear Register (GPIO32-GPIO38)
GPBTOGGLE	0x6FCE	2	GPIO B Toggle Register (GPIO32-GPIO38)
AIODAT	0x6FD8	2	Analog I/O Data Register (AIO0 - AIO15)
AIOSET	0x6FDA	2	Analog I/O Data Set Register (AIO0 - AIO15)
AIOCLEAR	0x6FDC	2	Analog I/O Clear Register (AIO0 - AIO15)
AIOTOGGLE	0x6FDE	2	Analog I/O Toggle Register (AIO0 - AIO15)

Основные настройки GPIO

Настройка выполняется через отдельные регистры группы «*GpioCtrlRegs*».

Регистр **GPxMUX** выбирает функцию вывода путём записи значения в соответствующее битовое поле:

- 0 – GPIO является вводом/выводом общего назначения и управляется программой
- 1/2/3 – GPIO является вводом/выводом периферийного устройства и управляется этим устройством

Регистр **GPxDIR** выбирает направление вывода, если он сконфигурирован как GPIO (GPxMUX = 0) путём записи значения в соответствующий бит:

- 0 – GPIO является вводом (можно только прочитать состояние GPIO)
- 1 – GPIO является выводом (можно изменить состояние GPIO)

Пример настройки GPIO


```
// Сконфигурировать GPIO0...6 как выводы модуля ШИМ
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 1;
GpioCtrlRegs.GPAMUX1.bit.GPIO1 = 1;
GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1;
GpioCtrlRegs.GPAMUX1.bit.GPIO3 = 1;
GpioCtrlRegs.GPAMUX1.bit.GPIO4 = 1;
GpioCtrlRegs.GPAMUX1.bit.GPIO5 = 1;

// Сконфигурировать GPIO34 как вывод
GpioCtrlRegs.GPBMUX1.bit.GPIO34 = 0;
GpioCtrlRegs.GPBDIR.bit.GPIO34 = 1;
```

Способы управления выводом

Управление осуществляется через отдельные регистры группы «*GpioDataRegs*».

Регистр **GPxSET** устанавливает на выводе выводе *высокий* уровень сигнала, при записи значения «1» в соответствующий бит.
Регистр **GPxCLEAR** устанавливает на выводе выводе *низкий* уровень сигнала, при записи значения «1» в соответствующий бит.
Регистр **GPxTOGGLE** *изменяет* уровень вывода при записи значения «1» в соответствующий бит: если вывод имел высокий уровень, то запись «1» в этот регистр переведёт его в низкий уровень. И наоборот – если вывод имел низкий уровень, то запись «1» в этот регистр переведёт его в высокий уровень.

Запись числа «0» в регистры **GPxSET и GPxCLEAR** не имеет никакого эффекта. При чтении эти регистры всегда имеют значение «0».

Регистр **GPxDAT** позволяет прочитать или изменить состояние GPIO:

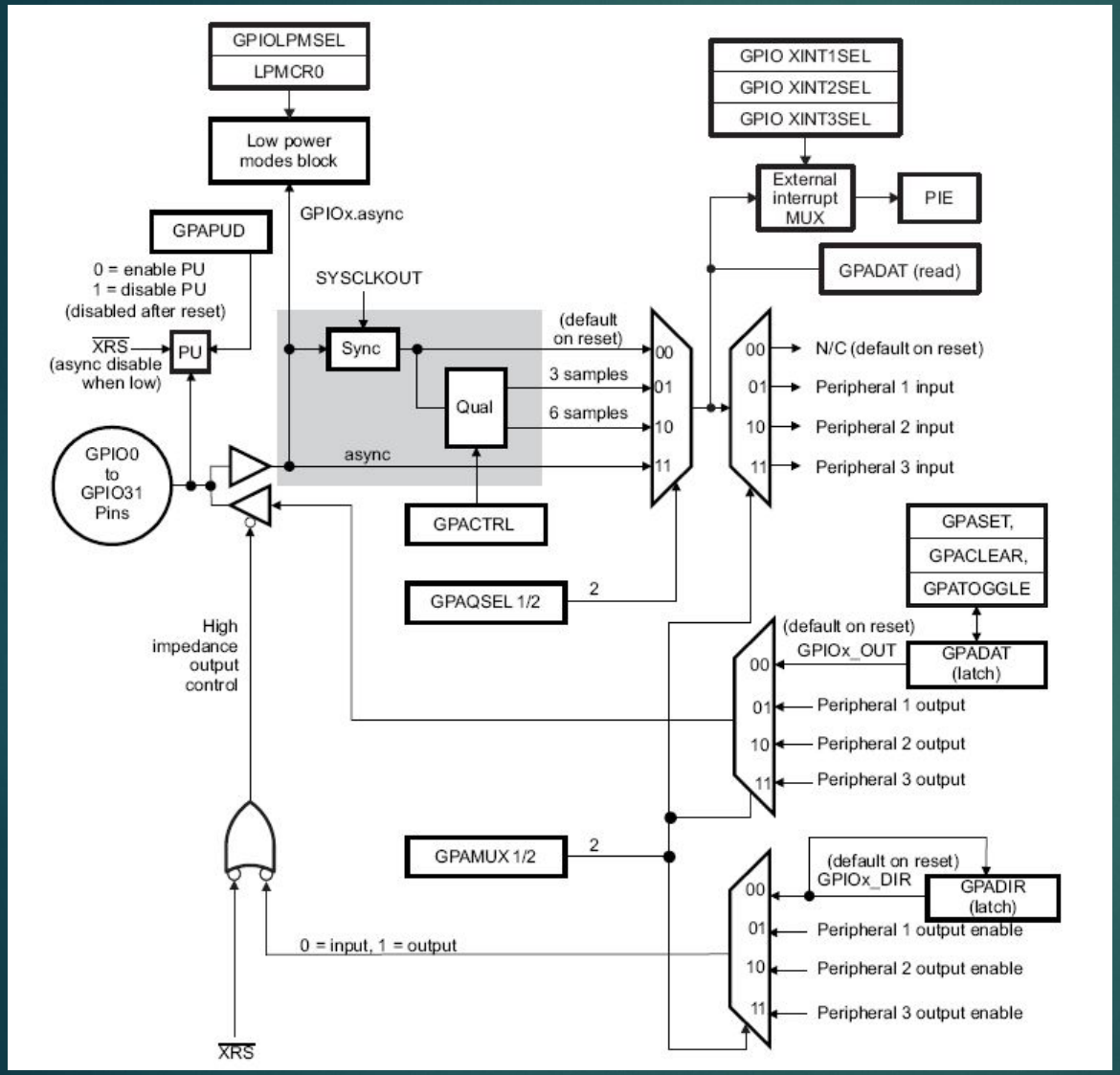
- Чтение: 0 означает, что GPIO имеет низкий сигнал (U_{GPIO} = 0V)
- Чтение: 1 означает, что GPIO имеет высокий сигнал (U_{GPIO} = 3,3V)
- Запись 0 или 1 присваивает соответствующий сигнал выводу, но только если он сконфигурирован как ввод/вывод общего назначения и является выводом через регистр (GPxMUX = 0 и GPxDIR = 1)

Пример управления выводом


```
// Функция включения реле
// Задаёт на GPIO34 высокий уровень сигнала
void relayOn (void) {
    GpioDataRegs.GPBSET.bit.GPIO34 = 1;
}

// Функция выключения реле
// Задаёт на GPIO34 низкий уровень сигнала
void relayOff (void) {
    GpioDataRegs.GPBCLEAR.bit.GPIO34 = 1;
}
```

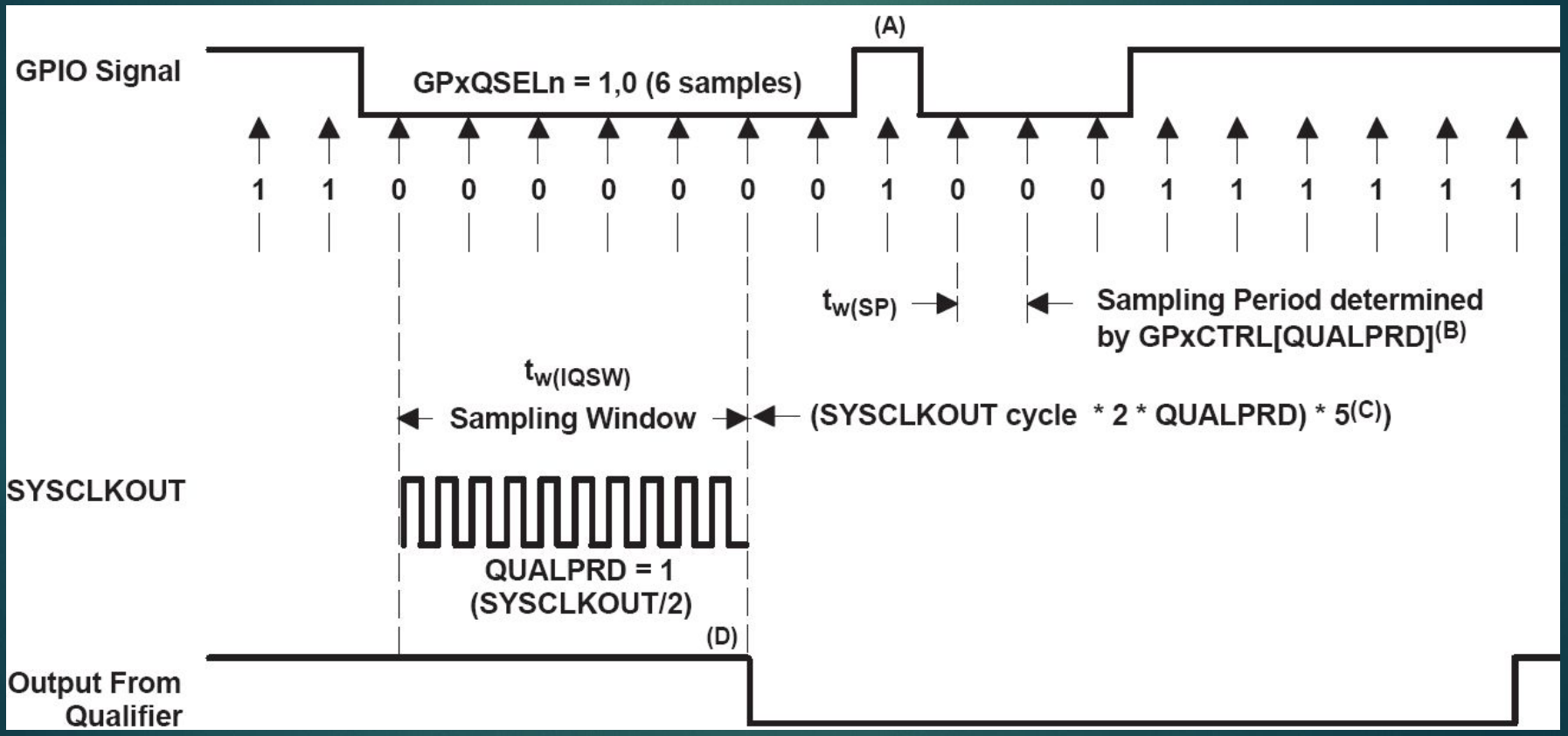
Фильтрация сигнала GPIO



- Все порты ввода/вывода могут быть предварительно зафильтрованы, чтобы исключить «дребезжание» сигнала
- Каждая «ножка» микроконтроллера может работать как ввод/вывод общего назначения или управляться периферийным модулем
- Сигнал вывода может быть синхронизирован с тактовой частотой процессора
- Выводы имеют настраиваемую внутреннюю привязку

Фильтрация сигнала GPIO

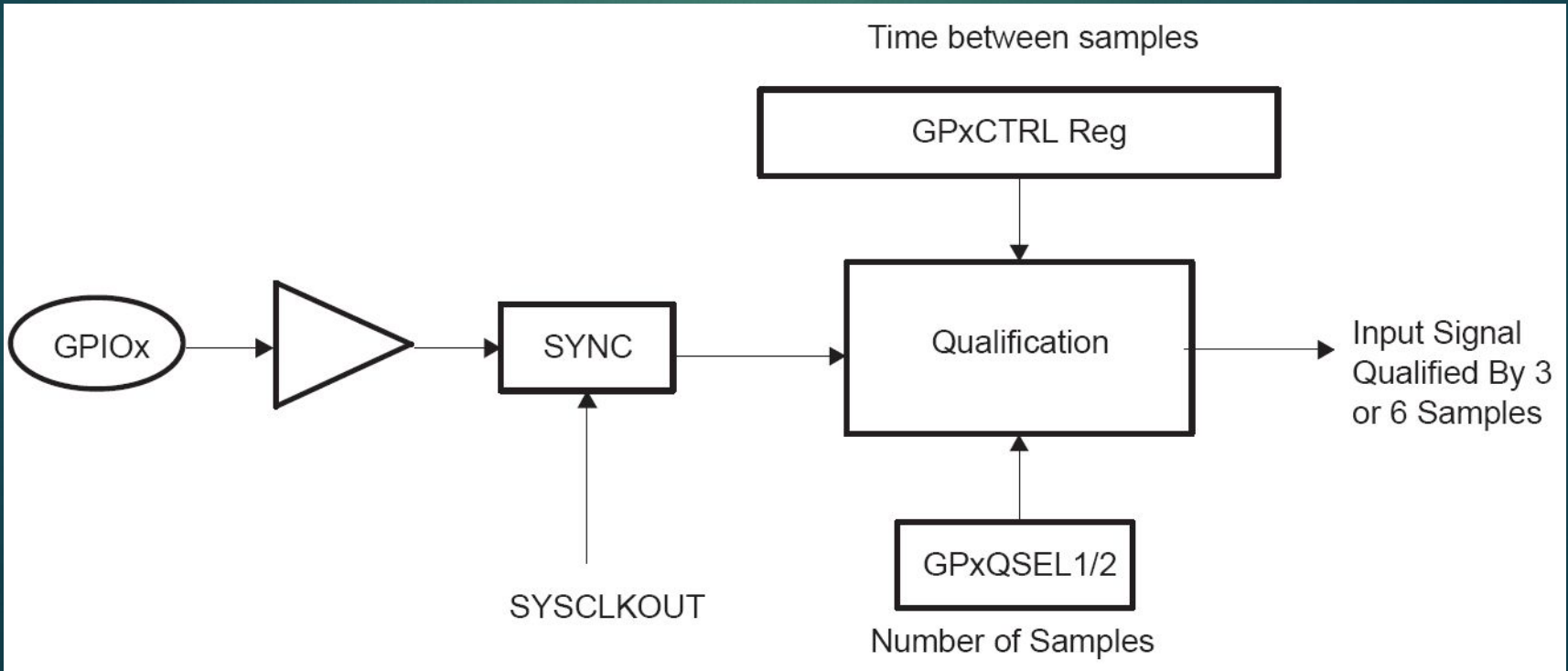
Чтобы избежать приём ложного сигнала (дребезг или помеха) можно настроить фильтрацию сигнала на GPIO. Например, считать сигнал истинным, только если он не меняет своего уровня в течение шести тактов.



Синхронизация сигнала GPIO

Для вводов GPIO можно настроить синхронизацию с тактовой частотой процессора: сигнал на входе может появиться в любой момент, однако можно настроить порты ввода так, чтобы приём сигналов осуществлялся синхронно с тактированием микроконтроллера, то есть сигнал на ножке проходит до процессора только на фронтах тактирующих сигналов.

Асинхронный приём сигнала возможен только если GPIO настроен настроены под определённую периферию



TMS320F28035 Менеджер прерываний (краткий обзор)

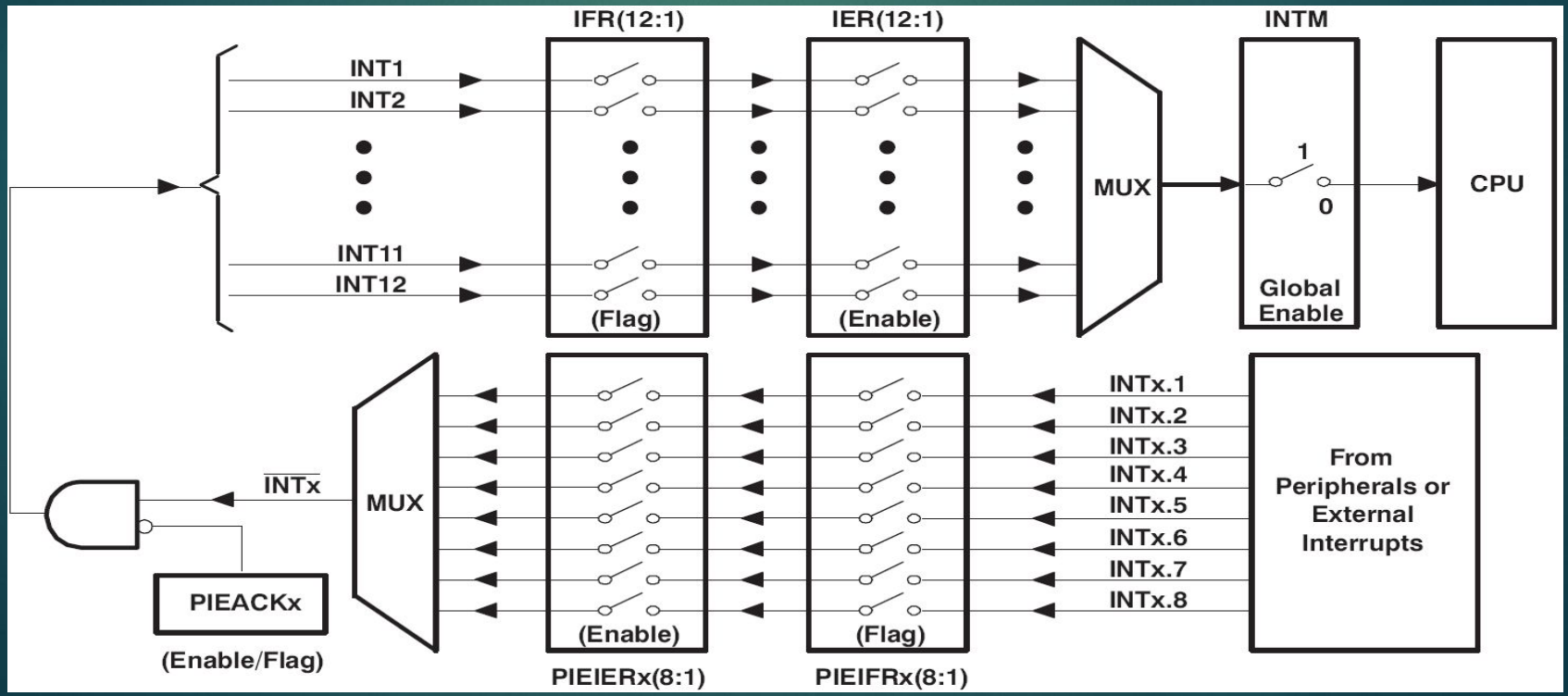
Таблица периферийных прерываний

	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1
INT1.y	WAKEINT	TINT0	ADCINT9	XINT2	XINT1	Reserved	ADCINT2	ADCINT1
	(LPM/WD)	(TIMER 0)	(ADC)	Ext. int. 2	Ext. int. 1	-	(ADC)	(ADC)
	0xD4E	0xD4C	0xD4A	0xD48	0xD46	0xD44	0xD42	0xD40
INT2.y	Reserved	EPWM7_TZINT	EPWM6_TZINT	EPWM5_TZINT	EPWM4_TZINT	EPWM3_TZINT	EPWM2_TZINT	EPWM1_TZINT
	–	(ePWM7)	(ePWM6)	(ePWM5)	(ePWM4)	(ePWM3)	(ePWM2)	(ePWM1)
	0xD5E	0xD5C	0xD5A	0xD58	0xD56	0xD54	0xD52	0xD50
INT3.y	Reserved	EPWM7_INT	EPWM6_ INT	EPWM5_INT	EPWM4_INT	EPWM3_INT	EPWM2_INT	EPWM1_INT
	-	(EPWM7)	(ePWM6)	(ePWM5)	(ePWM4)	(ePWM3)	(ePWM2)	(ePWM1)
	0xD6E	0xD6C	0xD6A	0xD68	0xD66	0xD64	0xD62	0xD60

INT9.y	Reserved	Reserved	ECANA_INT0	ECANA_INT1	LINA_INT1	LINA_INT0	SCITXINTA	SCIRXINTA
	–	–	(CAN-A)	(CAN-A)	(LIN-A)	(LIN-A)	(SCI-A)	(SCI-A)
	0xDCE	0xDCC	0xDCA	0xDC8	0xDC6	0xDC4	0xDC2	0xDC0
INT10.y	ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1
	(ADC)	(ADC)	(ADC)	(ADC)	(ADC)	(ADC)	(ADC)	(ADC)
	0xDDE	0xDDC	0xDDA	0xDD8	0xDD6	0xDD4	0xDD2	0xDD0
INT11.y	CLA1_INT8	CLA1_INT7	CLA1_INT6	CLA1_INT5	CLA1_INT4	CLA1_INT3	CLA1_INT2	CLA1_INT1
	(CLA)	(CLA)	(CLA)	(CLA)	(CLA)	(CLA)	(CLA)	(CLA)
	0xDEE	0xDEC	0xDEA	0xDE8	0xDE6	0xDE4	0xDE2	0xDE0
INT12.y	LUF	LVF	Reserved	Reserved	Reserved	Reserved	Reserved	XINT3
	(CLA)	(CLA)	–	–	–	–	–	Ext. Int. 3
	0xDFE	0xDFC	0xDFA	0xDF8	0xDF6	0xDF4	0xDF2	0xDF0

Мультиплицирование прерываний

При возникновении прерывания y в группе x, в регистре PIExIFR возводится соответствующий флаг. Если в регистре PIExIER установлен бит, разрешающий прерывание y в группе x, то в регистре процессора IFR взводится флаг соответствующей группы y. Если в регистре процессора IER установлен бит, разрешающий прерывание y, и регистр INTM глобально разрешает прерывания, то сигнал поступает на процессор и вызывается процедура обработки прерывания.



Источники процессорных прерываний