

# **ТЕМА 5: Архитектура микроконтроллера ATmega328P**

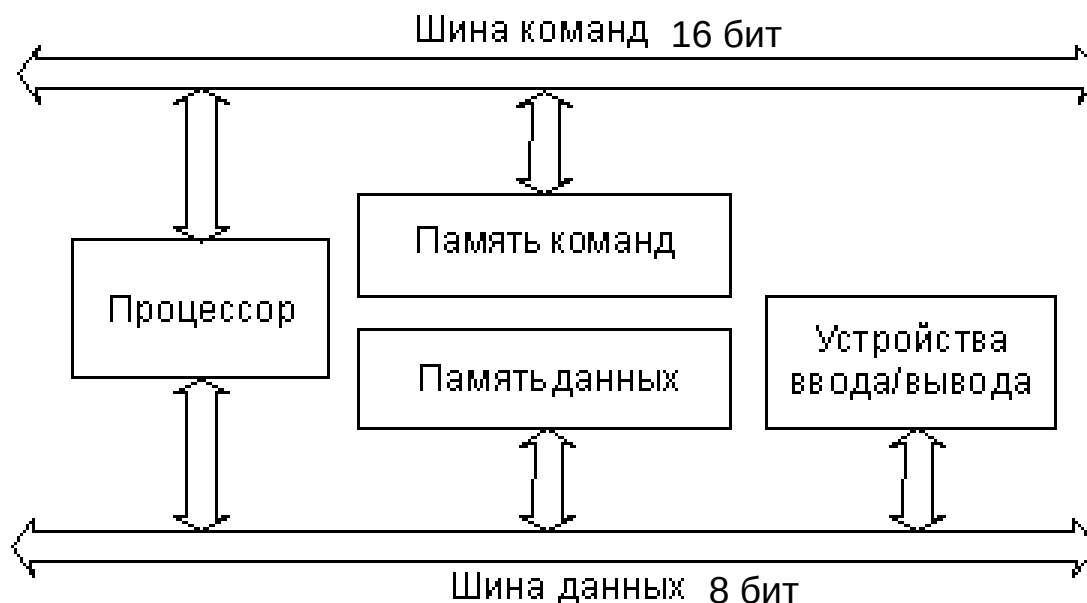
---

Содержание:

- 1. Общие сведения о микроконтроллере ATmega328P**
- 2. Порты ввода-вывода**

# Гарвардская архитектура

Микроконтроллер ATmega328P является 8 разрядным микроконтроллером, предназначенным для встраиваемых приложений. Он изготавливается по малопотребляющей КМОП технологии, которая в сочетании с усовершенствованной RISC архитектурой позволяет достичь наилучшего соотношения быстродействие/энергопотребление. Микроконтроллер построен по двухшинной (гарвардской) архитектуре и имеет отдельные шины памяти программ и памяти данных.



# Основные характеристики

---

- FLASH память программ объемом 32 Кбайт (число циклов стирания/записи не менее 1000);
- оперативная память (статическое ОЗУ) объемом 2 Кбайт;
- память данных на основе ЭСППЗУ (EEPROM) объемом 1 Кбайт (число циклов стирания/записи не менее 100000);
- возможность защиты от чтения и модификации памяти программ и данных;
- возможность программирования непосредственно в системе через последовательные интерфейсы SPI и JTAG;
- возможность самопрограммирования;
- возможность внутрисхемной отладки в соответствии со стандартом IEEE 1149.1 (JTAG);
- различные способы синхронизации: встроенный RC генератор с внутренней или внешней времязадающей RC цепочкой или с внешним резонатором (пьезокерамическим или кварцевым); внешний сигнал синхронизации;
- наличие нескольких режимов пониженного энергопотребления;
- наличие детектора снижения напряжения питания;
- возможность программного снижения частоты тактового генератора.

# ***Подсистема ввода-вывода***

---

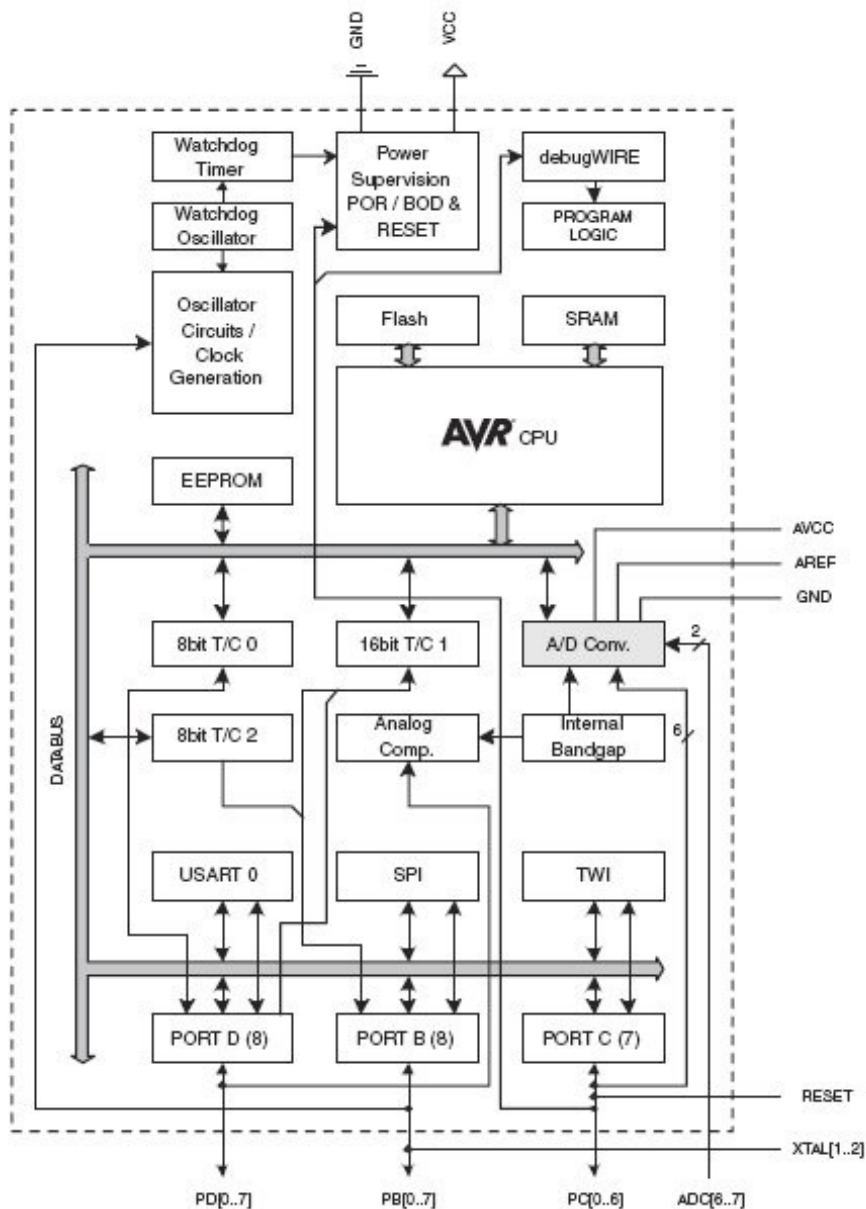
- 3 порта ввода-вывода (23 линии): В (8 линий), С (7 линий) и D (8 линий);
- программное конфигурирование и выбор портов ввода/вывода;
- выводы могут быть запрограммированы как входные или как выходные независимо друг от друга;
- входные буферы с триггером Шмитта на всех выводах;
- возможность подключения ко всем входам внутренних подтягивающих резисторов (сопротивление резисторов составляет 35...120 кОм).

# *Периферийные устройства*

---

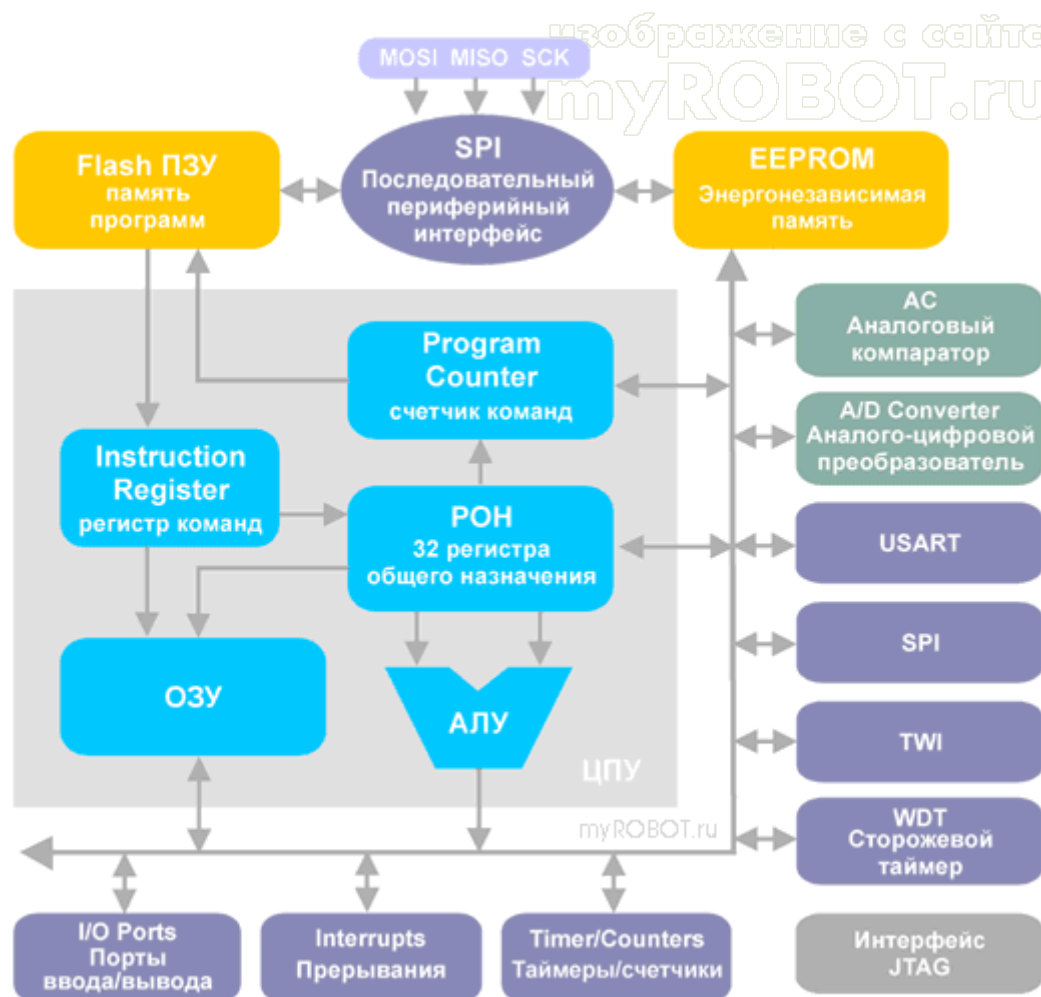
- 8 разрядные таймеры/счетчики (таймеры T0 и T2);
- 16 разрядный таймер/счетчик (таймер T1);
- сторожевой таймер WDT;
- 6 каналов ШИМ (широтно-импульсная модуляция);
- аналоговый компаратор;
- 6-ти канальный 10 разрядный АЦП;
- полнодуплексный универсальный синхронный/асинхронный приемопередатчик (USART);
- последовательный синхронный интерфейс SPI;
- последовательный двухпроводный интерфейс TWI (аналог интерфейса I2C).

# Структурная схема ATmega328P



# Архитектура ядра

Арифметико-логическое устройство (АЛУ), выполняющее все вычисления, подключено непосредственно к 32 рабочим регистрам, объединенным в регистровый файл.



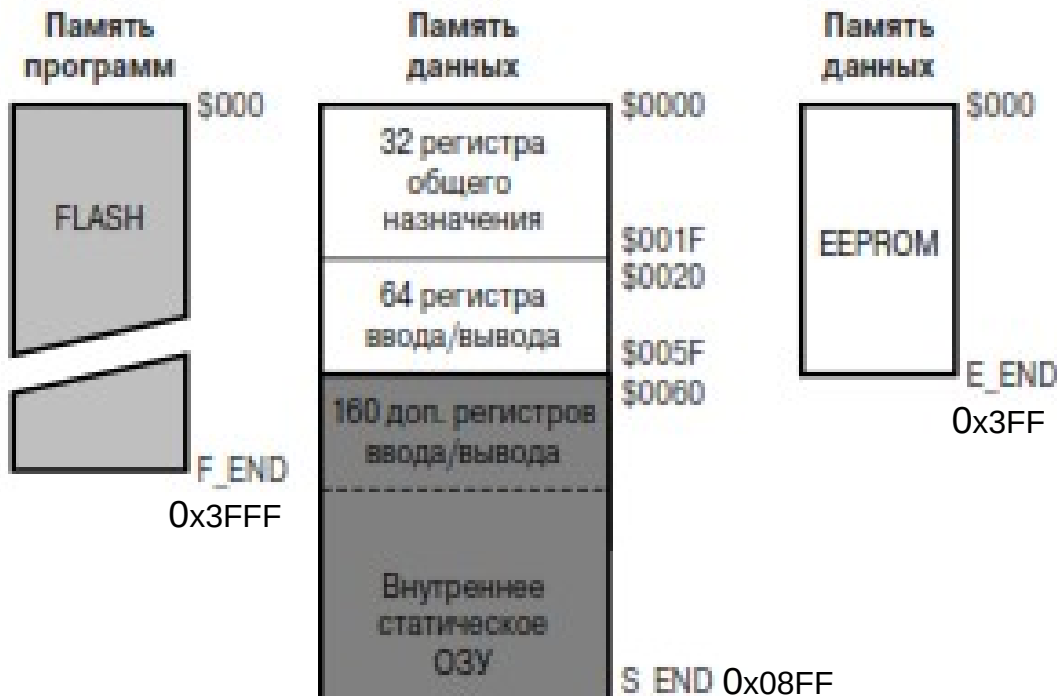
АЛУ выполняет одну операцию (чтение регистров, выполнение операции и запись результата в регистр) за один машинный цикл.

Практически каждая из команд (за исключением команд, у которых одним из операндов является 16 разрядный адрес) занимает одну ячейку памяти программ.

Конвейеризация заключается в том, что во время исполнения текущей команды производится выборка из памяти и дешиф-рация кода следующей команды.

# Организация памяти

В соответствии с Гарвардской архитектурой разделены не только адресные пространства памяти программ и памяти данных, но также и шины доступа к ним. Способы адресации и доступа к этим областям памяти также различны. Такая структура позволяет центральному процессору работать одновременно как с памятью программ, так и с памятью данных, что существенно увеличивает производительность.



Каждая из областей памяти данных (ОЗУ и EEPROM) также расположена в своем адресном пространстве.



# Память программ

---

Память программ предназначена для хранения команд, управляющих функционированием микроконтроллера. Память программ представляет собой электрически стираемое ППЗУ (FLASH ПЗУ).

Память программ имеет 16 разрядную организацию, поэтому для ATmega328P ее длина равна 16 К (16×1024) 16-ти разрядных слов.

Логически память программ разделена на две неравные части — область прикладной программы и область загрузчика (2 КБ). В последней может располагаться специальная программа (загрузчик), позволяющая микроконтроллеру самостоятельно управлять загрузкой и выгрузкой прикладных программ.

Для адресации памяти программ используется 16-ти разрядный счетчик команд (Program Counter).

По адресу \$0000 памяти программ находится вектор сброса. После инициализации (сброса) микроконтроллера выполнение программы начинается с этого адреса (по этому адресу должна размещаться команда перехода к инициализационной части программы). Начиная с адреса \$0002 располагается таблица векторов прерываний.

# Память данных

---

Память данных микроконтроллеров семейства Мега разделена на три части:

- регистровая память,
- оперативная память (статическое ОЗУ)
- энергонезависимое ЭСППЗУ (EEPROM).

**Регистровая память** включает:

- 32 регистра общего назначения (РОН), объединенных в файл,
- служебные регистры ввода/вывода (РВВ) и дополнительные регистры ввода-вывода (ДРВВ). Под РВВ в памяти микроконтроллера отводится 64 байта, а под ДРВВ – 160 байт.

**Оперативная память** (статическое ОЗУ) объемом 2 Кбайт служит для хранения переменных программ помимо регистров общего назначения.

**Энергонезависимая постоянная память** служит для долговременного хранения различной информации, которая может изменяться в процессе функционирования готовой системы (калибровочные константы, серийные номера, ключи и т. п.). Ее объем 1 Кбайт. Эта память расположена в отдельном адресном пространстве, а доступ к ней осуществляется с помощью специальных регистров ввода-вывода (РВВ).

# Статическое ОЗУ

Используется линейная организация памяти. Первые 256 ячеек отведены под регистры: 32 штуки регистры общего назначения, остальное – регистры ввода-вывода (РВВ и ДРВВ).

## Data Memory

32 Registers	0x0000 - 0x001F
64 I/O Registers	0x0020 - 0x005F
160 Ext I/O Reg.	0x0060 - 0x00FF
Internal SRAM (512/1024/1024/2048 x 8)	0x0100  0x08FF

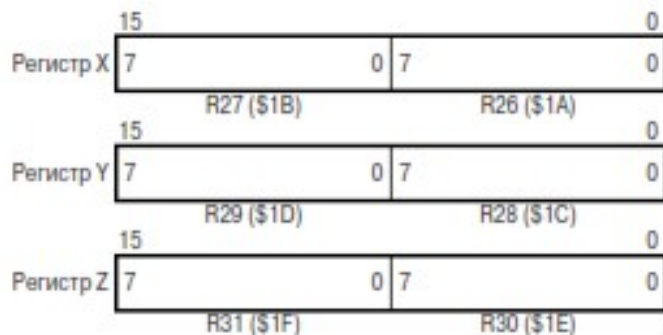
# Регистры общего назначения

Все регистры общего назначения объединены в регистровый файл быстрого доступа.

Все 32 РОН непосредственно доступны АЛУ. Любой РОН может использоваться практически во всех командах и как операнд источник и как операнд приемник. Это позволяет АЛУ выполнять одну операцию (извлечение операндов из регистрового файла, выполнение команды и запись результата обратно в регистровый файл) за один машинный цикл.

Каждый регистр файла имеет свой собственный адрес в пространстве памяти данных. Поэтому к ним можно обращаться двумя способами (как к регистрам и как к памяти).

Последние 6 регистров файла (R26...R31) могут также объединяться в три 16 разрядных регистра X, Y и Z, используемых в качестве указателей при косвенной адресации памяти данных.



7	0	Адрес
R0	\$00	
R1	\$01	
R2	\$02	
...		
R13	\$0D	
R14	\$0E	
R15	\$0F	
R16	\$10	
R17	\$11	
...		
R26	\$1A	регистр X, мл.байт
R27	\$1B	регистр X, ст.байт
R28	\$1C	регистр Y, мл.байт
R29	\$1D	регистр Y, ст.байт
R30	\$1E	регистр Z, мл.байт
R31	\$1F	регистр Z, ст.байт

# Регистры ввода-вывода

Все регистры ввода/вывода (РВВ) условно можно разделить на две группы:

- служебные регистры микроконтроллера
- регистры, относящиеся к конкретным периферийным устройствам.

Регистры ввода/вывода располагаются в так называемом основном пространстве ввода/вывода, в котором их адреса начинаются с 0x00 до 0x3F. При этом по указанным адресам к ним можно обратиться с помощью команд IN и OUT.

Однако эти регистры также представлены в общем адресном пространстве статической памяти как ячейки ОЗУ, где их адреса начинаются с 0x20 до 0x5F (т.е. сдвинуты на константу 0x20). Как к ячейкам ОЗУ к регистрам можно обратиться с помощью команд ST/SD/SDD и LD/LDS/LDD.

**К дополнительным регистрам ввода-вывода (ДРВВ) можно обращаться только как к ячейкам памяти.**

Регистровый файл		Адресное пространство
R0		\$0000
R1		\$0001
R2		\$0002
...		...
R31		\$001F

Регистры ввода/вывода

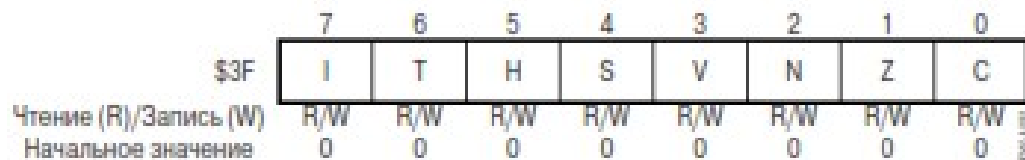
\$00		\$0020
\$01		\$0021
...		...
\$3F		\$005F

Доп. регистры ввода/вывода

\$60		\$0060
\$61		\$0061
...		...
\$FF		\$00FF

# Регистр состояния SREG

SREG располагается по адресу \$3F (\$5F) и содержит набор флагов, показывающих текущее состояние микроконтроллера. Большинство флагов автоматически устанавливаются в «1» или сбрасываются в «0» при наступлении определенных событий:



Разряд	Название	Описание
7	I	Общее разрешение прерываний. Для разрешения прерываний этот флаг должен быть установлен в «1». Разрешение/запрещение отдельных прерываний производится установкой или сбросом соответствующих разрядов регистров масок прерываний (регистра управления прерываниями). Если флаг сброшен, то прерывания запрещены независимо от состояния разрядов этих регистров. Флаг сбрасывается аппаратно после входа в прерывание и восстанавливается командой RETI для разрешения обработки следующих прерываний
6	T	Хранение копируемого бита. Этот разряд регистра используется в качестве источника или приемника команд копирования битов BLD (Bit Load) и BST (Bit Store). Заданный разряд любого ПОН может быть скопирован в этот разряд командой BST или установлен в соответствии с содержимым данного разряда командой BLD

## Регистр состояния SREG (продолжение)

Раз- ряд	Наз- вание	Описание
5	N	Флаг половинного переноса. Этот флаг устанавливается в «1», если произошел перенос из младшей половины байта (из 3-го разряда в 4-й) или заем из старшей половины байта при выполнении некоторых арифметических операций
4	S	Флаг знака. Этот флаг равен результату операции «Исключающее ИЛИ» (XOR) между флагами N (отрицательный результат) и V (переполнение числа в дополнительном коде). Соответственно этот флаг устанавливается в «1», если результат выполнения арифметической операции меньше нуля
3	V	Флаг переполнения дополнительного кода. Этот флаг устанавливается в «1» при переполнении разрядной сетки знакового результата. Используется при работе со знаковыми числами (представленными в дополнительном коде). Более подробно — см. описание системы команд
2	N	Флаг отрицательного значения. Этот флаг устанавливается в «1», если старший (7-й) разряд результата операции равен «1». В противном случае флаг равен «0»
1	Z	Флаг нуля. Этот флаг устанавливается в «1», если результат выполнения операции равен нулю
0	C	Флаг переноса. Этот флаг устанавливается в «1», если в результате выполнения операции произошел выход за границы байта



## Содержание:

1. **Общие сведения о микроконтроллере ATmega328P**
2. Порты ввода-вывода



# Порты ввода-вывода

---

Каждый порт микроконтроллеров состоит из определенного числа выводов, через которые микроконтроллер может осуществлять прием и передачу цифровых сигналов. Задание направления передачи данных через любой контакт ввода/вывода может быть произведено программно в любой момент времени.

Микроконтроллер ATmega8x имеют три порта ввода/вывода:

- порт В (8 разрядный),
- порт С (7 разрядный),
- порт D (8 разрядный).

Всего контактов ввода/вывода.

Обращение к портам производится через регистры ввода/вывода. Под каждый порт в адресном пространстве ввода/вывода зарезервировано по 3 адреса, по которым размещены следующие регистры:

- регистр данных порта PORTx,
- регистр направления данных DDRx ,
- регистр выводов порта PINx.

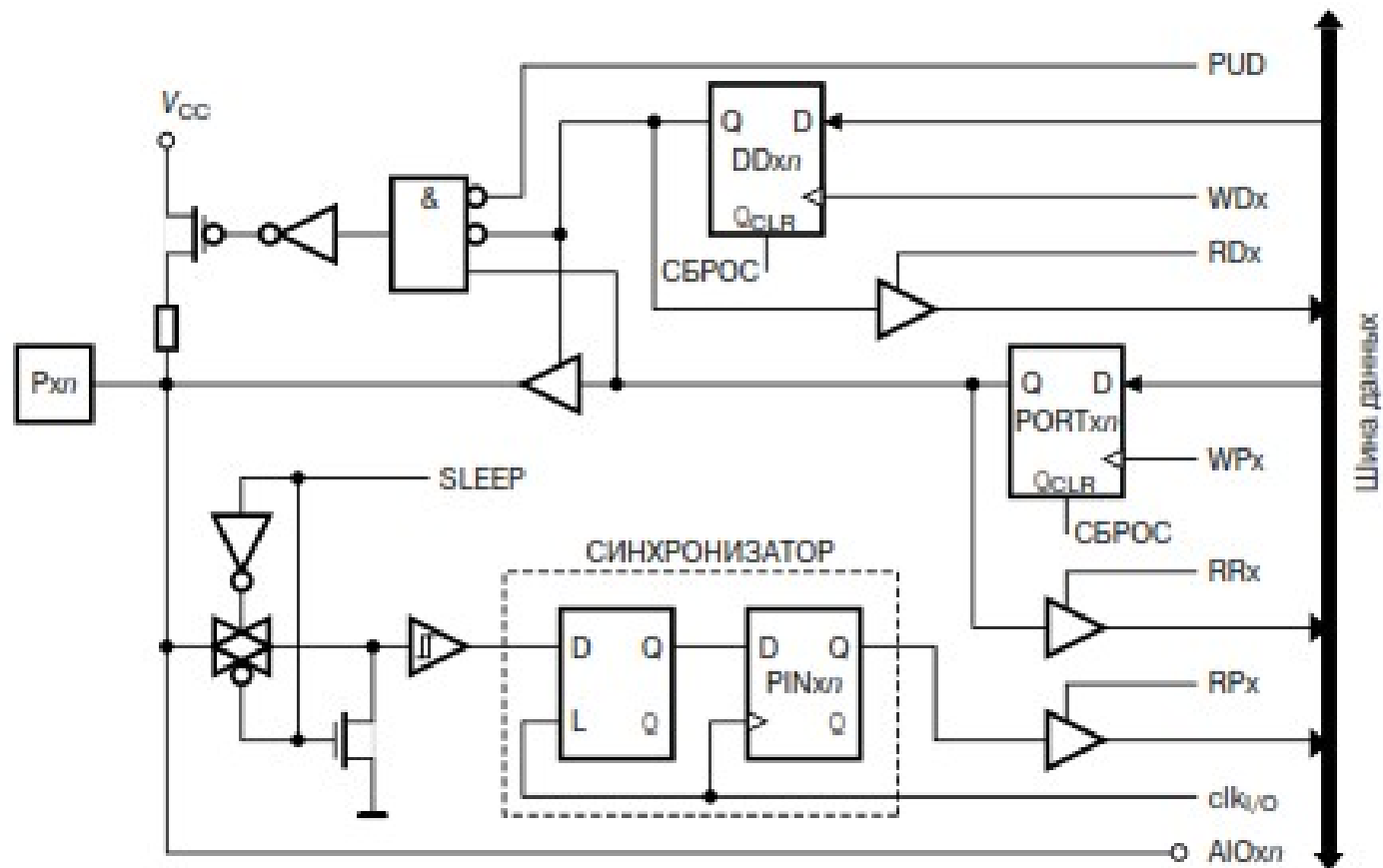
Регистр PINx доступен только для чтения, а PORTx и DDRx доступны как для чтения, так и для записи.

# Порты ввода-вывода

В таблице приведены адреса регистров микроконтроллера ATmega328P.

Порт	Регистр	Адреса
B	PORTB	\$18 (\$38)
	DDRB	\$17 (\$37)
	PINB	\$16 (\$36)
C	PORTC	\$15 (\$38)
	DDRC	\$14 (\$37)
	PINC	\$13 (\$36)
D	PORTD	\$12 (\$32)
	DDRD	\$11 (\$31)
	PIND	\$10 (\$30)

# Схема линии порта ввода-вывода



**PUD:** выключение подт. резистора  
**SLEEP:** управление спящим режимом  
 **$clk_{I/O}$ :** тактовый сигнал подсистемы ввода/вывода

**$WDr$ :** запись регистра  $DDR_x$   
 **$RDx$ :** чтение регистра  $DDR_x$   
 **$WPx$ :** запись регистра  $PORT_x$   
 **$RRx$ :** чтение регистра  $PORT_x$   
 **$RPx$ :** чтение выводов порта  $PORT_x$

**$AIO_{xn}$ :** аналоговый вход/выход вывода  $n$  порта  $PORT_x$

# Конфигурирование портов ввода-вывода

Разряд **DDxn** регистра DDx определяет направление передачи данных через контакт ввода/вывода. Если этот разряд установлен в «1», то *n* й вывод порта является выходом, если же сброшен в «0» — входом.

Разряд **PORTxn** регистра PORTx выполняет двойную функцию.

1. Если вывод функционирует как выход ( $DDxn = \text{«1»}$ ), этот разряд определяет состояние вывода порта. Если разряд установлен в «1», на выводе устанавливается напряжение ВЫСОКОГО уровня. Если разряд сброшен в «0», на выводе устанавливается напряжение НИЗКОГО уровня.

2. Если вывод функционирует как вход ( $DDxn = \text{«0»}$ ), разряд **PORTxn** определяет состояние внутреннего подтягивающего резистора для данного вывода. При установке разряда **PORTxn** в «1» подтягивающий резистор подключается между выводом микроконтроллера и проводом питания.

DDRxn	PORTxn	Состояние линии	Описание
0	0	I (Input) Вход	Высокоимпедансный вход. (Не рекомендую использовать, так как могут наводиться наводки от питания)
0	1	I (Input) Вход	Подтянуто внутренне сопротивление
1	0	O (Output) Выход	На выходе низкий уровень
1	1	O (Output) Выход	На выходе высокий уровень