

## CHƯƠNG 2 KIẾN TRÚC PHẦN CỨNG HỌ AVR ATMEL - ATmega324P

### 2.1 TỔNG QUAN HỌ AVR

AVR cơ bản được thiết kế bởi 2 sinh viên của Học viện công nghệ Na Uy (Norwegian Institute of Technology) là Alf-Egil Bogen và Vegard Wollan. Sau đó được công ty Atmel mua lại và phát triển vào năm 1996. AVR là vi điều khiển 8 bit dạng RISC (Reduced Instruction Set Computer) theo kiến trúc Harvard. Ngoại trừ AVR32 là vi điều khiển 32 bit. Tên gọi AVR không có khái niệm nhất định, nhưng thường được hiểu là viết tắt tên của 2 người thiết kế: AVR - Alf and Vegard RISC.

#### 2.1.1 Đặc tính chung AVR

- Bộ nhớ chương trình ROM: không gian bộ nhớ có thể lên tới 8MB, tuy nhiên không phải tất cả đều được đạt mức đó. Khi ghi ROM, kích thước của ROM có thể từ 1KB đến 256KB tùy theo loại. AVR sử dụng bộ nhớ Flash on-chip có thể xóa nhanh trong vài giây.
- Bộ nhớ dữ liệu gồm RAM và EEPROM: AVR có không gian RAM tối đa là 64KB, tùy theo loại. Bộ nhớ RAM chia thành 3 phần: các thanh ghi làm việc đa dụng (gereral purpose working registers), bộ nhớ I/O (I/O memory), và bộ nhớ SRAM bên trong. Tất cả các loại AVR đều có 32 thanh ghi làm việc đa dụng; nhưng sẽ khác nhau về kích thước của SRAM và bộ nhớ I/O. AVR dùng EEPROM chứa các dữ liệu cố định.
- Các chân I/O: có từ 3 đến 86 chân để xuất/nhập tùy loại.
- Ngoài vi: các ngoại vi bao gồm bộ biến đổi ADC, bộ định thời và các loại giao tiếp nối tiếp USART (Universal Synchronous Asynchronous Receiver Transmitter), I2C (TWI) và SPI. Ngoài ra mỗi loại có thêm ngoại vi cho nhiệm vụ riêng.

#### 2.1.2 Các nhóm trong họ vi điều khiển AVR

Họ AVR chia thành 4 nhóm: Classic, Mega, Tiny và nhóm mục đích đặc biệt (special purpose).

- Classic AVR (AT90Sxxxx): là nhóm vi điều khiển gốc, hiện không còn được sử dụng.
- Mega AVR (ATmegaxxxx): là nhóm vi điều khiển có hơn 120 lệnh và nhiều ngoại vi khác nhau. Do vậy nhóm này có nhiều ứng dụng rộng rãi cho các thiết kế khác nhau.

Bảng 2.1 Các chip tiêu biểu trong nhóm ATmega

IC	ROM chương trình	RAM dữ liệu	EEPROM dữ liệu	Số chân I/O	ADC	Bộ định thời	Số chân Loại vỏ
<b>ATmega8</b>	8K	1K	0.5K	23	8	3	TQFP32, PDIP28
<b>ATmega16</b>	16K	1K	0.5K	32	8	3	TQFP44, PDIP40
<b>ATmega32</b>	32K	2K	1K	32	8	3	TQFP44, PDIP40
<b>ATmega64</b>	64K	4K	2K	54	8	4	TQFP64, MLF64
<b>Atmega1280</b>	128K	8K	4K	86	16	6	TQFP100, CBGA

- Tiny AVR (ATtinyxxxx): có ít lệnh và kích thước nhỏ so với ATmega.
- AVR cho mục đích đặc biệt: các IC này cũng thuộc các nhóm trên nhưng được trang bị thêm để thiết kế ứng dụng cụ thể. Ví dụ như bộ điều khiển USB, bộ điều khiển LCD, bộ điều khiển mạng cục bộ, ...

Các IC AVR có mã số bắt đầu là chữ AT (Atmel) và tên nhóm, theo sau là dãy số. Hầu hết các trường hợp theo dãy số này có thể cho biết dung lượng ROM của AVR. Quy tắc là từ trái sang phải, lấy số có giá trị lớn nhất có dạng  $2^n$ .

**Ví dụ 2.1:** ATmega324 thì giá trị đúng theo quy tắc là 32, vậy có 32KB ROM.  
ATtiny44 thì có 4KB ROM

Trong giáo trình này, chúng ta sử dụng ATmega324P do có nhiều ứng dụng rộng rãi. Ngoài ra nó có dạng PDIP dễ dàng thực hiện việc lắp ráp mạch thực tế.

### Câu hỏi ôn tập

1. Cho biết tên các nhóm thuộc họ AVR.
2. Nhóm nào có kích thước nhỏ nhất.
3. Cho biết kích thước ROM chương trình của:  
(a) ATtiny25    (b) ATmega324    (c) ATmega1280
4. Cho biết kích thước RAM của:  
(a) ATmega8    (b) ATmega32    (c) ATmega1280

## 2.2 VI ĐIỀU KHIỂN ATmega324P

### 2.2.1 Đặc tính kỹ thuật

ATmega324P là vi điều khiển 8 bit CMOS công suất thấp theo kiến trúc RISC nâng cao AVR. Bằng việc thực hiện lệnh chỉ trong 1 chu kỳ xung nhịp, vi điều khiển có thể đạt được tốc độ 1 MIPS (triệu lệnh mỗi giây) theo từng MHz.

- **Kiến trúc RISC nâng cao**
  - Có 131 lệnh.
  - Đa số các lệnh thực thi 1 chu kỳ xung nhịp.
  - 32 thanh ghi làm việc đa dụng 8 bit.
  - Có thể đạt tới 20 MIPS với tần số 20MHz.
  - Bộ nhân 2 chu kỳ on-chip
- **Các thành phần bộ nhớ**
  - Bộ nhớ chương trình Flash 32KB
  - Bộ nhớ EEPROM 1KB
  - Bộ nhớ SRAM 2KB
- **Các ngoại vi**
  - 32 đường I/O lập trình được
  - 2 bộ định thời/bộ đếm 8 bit với bộ chia tần riêng biệt và chế độ so sánh
  - 1 bộ định thời/bộ đếm 16 bit với bộ chia tần riêng biệt, chế độ so sánh và chế độ capture.
  - Bộ đếm thời gian thực với bộ dao động riêng biệt.
  - 6 kênh điều chế độ rộng xung PWM.

- 8 kênh ADC 10 bit, ngõ vào đơn, ngõ vào vi sai lập trình được độ lợi (PGA)
- Giao tiếp nối tiếp 2 dây I2C.
- 2 bộ USART lập trình được.
- Giao tiếp nối tiếp SPI.
- Bộ định thời watchdog lập trình được với bộ dao động on-chip riêng biệt.
- Bộ so sánh tương tự on-chip.
- Ngắt và wake-up trên các chân.

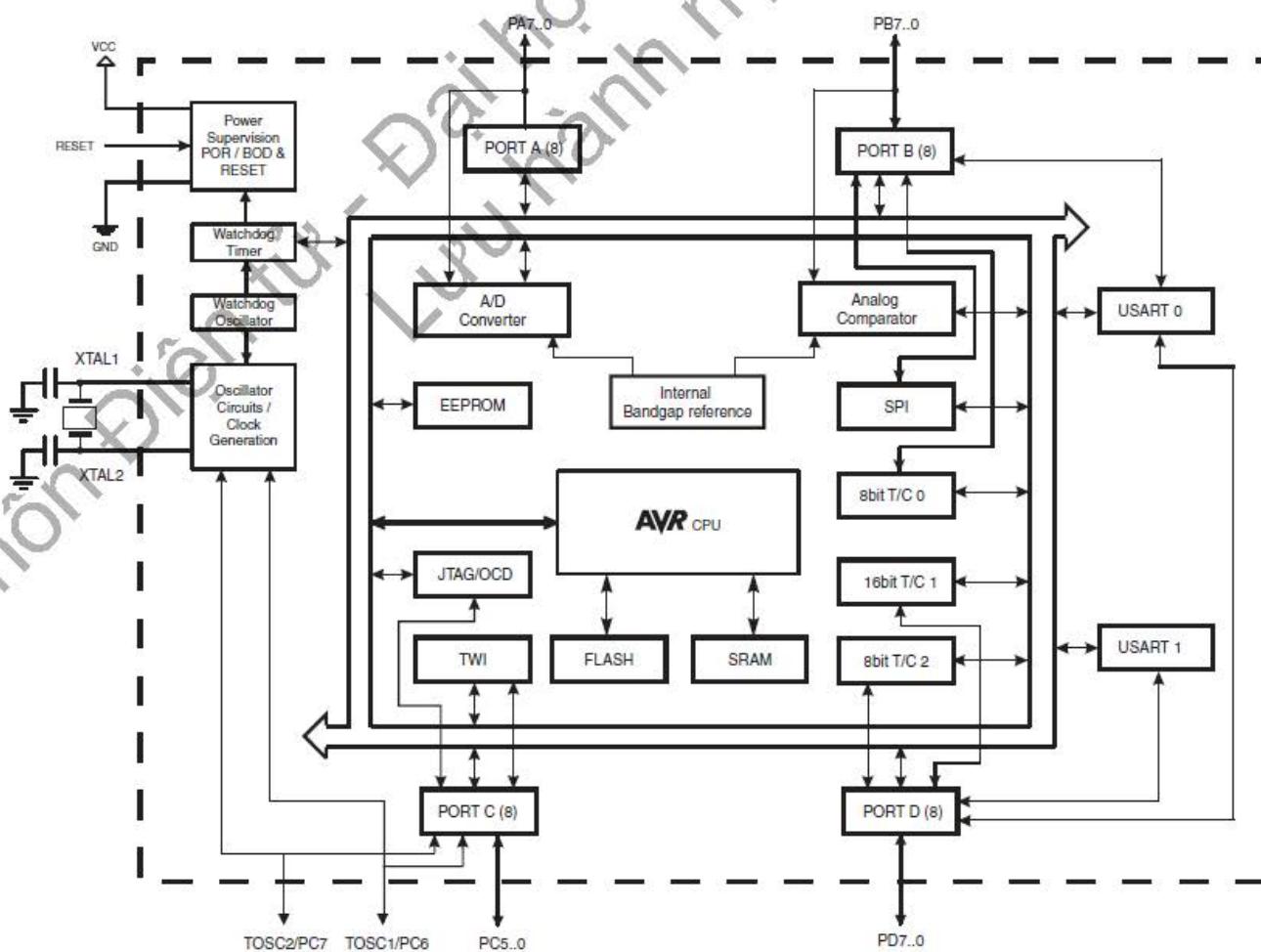
- **Các tính năng đặc biệt**

- Mạch khởi động Power-on và bộ phát hiện sụt áp nguồn Brown-out lập trình được.
- Bộ dao động RC điều chỉnh bên trong.
- Có nhiều nguồn ngắt bên trong và bên ngoài.
- Có 6 chế độ ngủ tiết kiệm năng lượng nguồn.

- **Đặc tính khác**

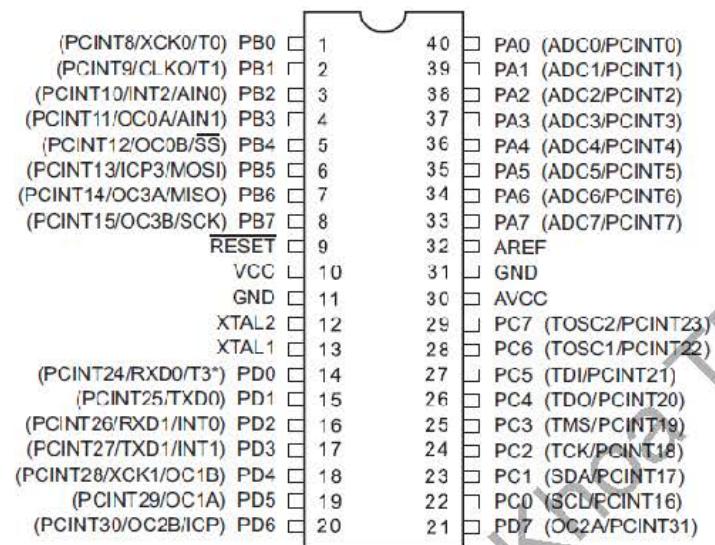
- Có các dạng đóng gói PDIP 40 chân, TQFP 44 chân và VQFN/QFN/MLF 44 chân.
- Điện áp hoạt động 2.7V – 5.5V.
- Tần số làm việc 0 – 20MHz.
- JTAG chuẩn IEEE 7149.1 giúp nạp chương trình trực tiếp trên hệ thống đang sử dụng.

### 2.2.2 Sơ đồ khái

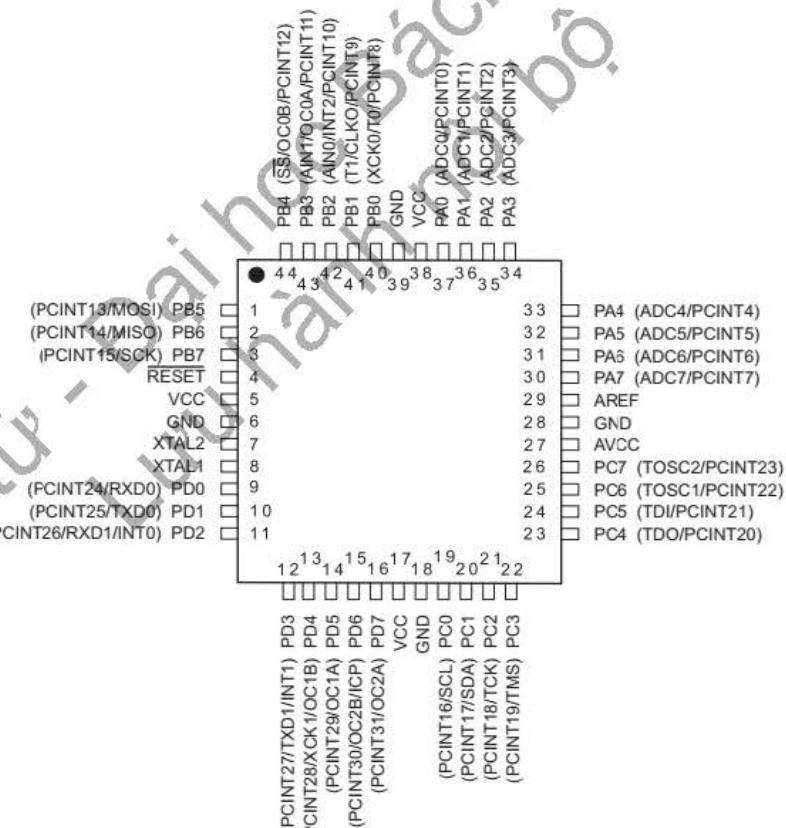


Hình 2.1 Sơ đồ khái AVR

### 2.2.3 Sơ đồ chân



(a) Dạng vỏ PDIP



(b) Dạng vỏ TQFP/VQFN/QFL/MLF

Hình 2.2 Sơ đồ chân ATmega324P

- **VCC:** điện áp nguồn.

Là chân cung cấp điện áp nguồn cho chip. Điện áp nguồn từ 2.7V đến 5.5V.

- **GND:** chân nối đất

### - RESET

Ngõ vào reset. Khi chân này ở mức thấp dài hơn độ dài xung tối thiểu thì sẽ reset MCU.

### - XTAL1, XTAL2

Chân kết nối với mạch dao động thạch anh hay mạch dao động RC tạo xung clock hoạt động cho AVR. Khi dùng mạch xung nhịp ngoài thì nối vào chân XTAL1.

### - AVCC

Là chân điện áp nguồn cho bộ ADC.

### - AREF

Là chân điện áp tham chiếu analog cho bộ ADC.

### - Port A (PA7:PA0)

Port A là port I/O 8 bit 2 chiều có điện trở kéo lên bên trong, có thể lập trình riêng cho từng bit. Khi reset các chân port1 là 3 trạng thái.

Port A cũng có chức năng khác như sau:

Bảng 2.2 Các chức năng khác của chân Port A

Chân port	Chức năng luân phiên
PA7	ADC7 (ADC input channel 7) PCINT7 (Pin Change Interrupt 7)
PA6	ADC6 (ADC input channel 6) PCINT6 (Pin Change Interrupt 6)
PA5	ADC5 (ADC input channel 5) PCINT5 (Pin Change Interrupt 5)
PA4	ADC4 (ADC input channel 4) PCINT4 (Pin Change Interrupt 4)
PA3	ADC3 (ADC input channel 3) PCINT3 (Pin Change Interrupt 3)
PA2	ADC2 (ADC input channel 2) PCINT2 (Pin Change Interrupt 2)
PA1	ADC1 (ADC input channel 1) PCINT1 (Pin Change Interrupt 1)
PA0	ADC0 (ADC input channel 0) PCINT0 (Pin Change Interrupt 0)

ADC7:0 có chức năng làm 8 kênh ngõ vào của bộ ADC

PCINT7:0 có chức năng làm 8 nguồn ngắt Pin Change.

### - Port B (PB7:PB0)

Port B là port I/O 8 bit 2 chiều có điện trở kéo lên bên trong, có thể lập trình riêng cho từng bit. Khi reset các chân port1 là 3 trạng thái.

Port B cũng thực hiện các chức năng khác như sau:

**Bảng 2.3 Các chức năng khác của chân Port B**

Chân port	Chức năng luân phiên
PB7	SCK (SPI Bus Master clock input) PCINT15 (Pin Change Interrupt 15)
PB6	MISO (SPI Bus Master Input/Slave Output) PCINT14 (Pin Change Interrupt 14)
PB5	MOSI (SPI Bus Master Output/Slave Input) PCINT13 (Pin Change Interrupt 13)
PB4	SS (SPI Slave Select input) OC0B (Timer/Counter 0 Output Compare Match B Output) PCINT12 (Pin Change Interrupt 12)
PB3	AIN1 (Analog Comparator Negative Input) OC0A (Timer/Counter 0 Output Compare Match A Output) PCINT11 (Pin Change Interrupt 11)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input) PCINT10 (Pin Change Interrupt 10)
PB1	T1 (Timer/Counter 1 External Counter Input) CLKO (Divided System Clock Output) PCINT9 (Pin Change Interrupt 9)
PB0	T0 (Timer/Counter 0 External Counter Input) XCK0 (USART0 External Clock Input/Output) PCINT8 (Pin Change Interrupt 8)

- **PB7: SCK/PCINT15**

**SCK:** là ngõ ra xung clock khi làm master và là ngõ vào xung clock khi làm slave trong giao tiếp SPI.

**PCINT15:** nguồn ngắt Pin Change 15.

- **PB6: MISO/PCINT14**

**MISO:** là ngõ vào dữ liệu khi làm master và là ngõ ra dữ liệu khi làm slave trong giao tiếp SPI.

**PCINT14:** nguồn ngắt Pin Change 14.

- **PB5: MOSI/PCINT13**

**MOSI:** là ngõ ra dữ liệu khi làm master và là ngõ vào dữ liệu khi làm slave trong giao tiếp SPI.

**PCINT13:** nguồn ngắt Pin Change 13.

- **PB4: SS/OC0B/PCINT12**

**SS:** ngõ vào lựa chọn cổng slave dùng trong giao tiếp SPI.

**OC0B:** ngõ ra so sánh ngõ ra Match B của Bộ định thời/Bộ đếm 0. Chân này cũng là ngõ ra khi bộ định thời ở chế độ PWM.

**PCINT12:** nguồn ngắt Pin Change 12.

- **PB3: AIN1/OC0A/PCINT11**

**AIN1:** ngõ vào âm của bộ so sánh tương tự (analog comparator).

**OC0A:** ngõ ra so sánh ngõ ra Match A của Bộ định thời/Bộ đếm 0. Chân này cũng là ngõ ra khi bộ định thời ở chế độ PWM.

**PCINT11:** nguồn ngắt Pin Change 11.

- **PB2: AIN0/INT2/PCINT10**

AIN0: ngõ vào dương của bộ so sánh tương tự (analog comparator).

INT2: ngắt ngoài 2.

PCINT10: nguồn ngắt Pin Change 10.

- **PB1: T1/CLKO/PCINT9**

T1: ngõ vào bộ đếm của Bộ định thi/Bộ đếm 1.

CLKO: ngõ ra từ bộ chia xung nhịp.

PCINT9: nguồn ngắt Pin Change 9.

- **PB0: T0/XCK0/PCINT8**

T0: ngõ vào bộ đếm của Bộ định thi/Bộ đếm 0.

XCK0: xung nhịp bên ngoài cho USART 0 khi dùng ở chế độ đồng bộ.

PCINT8: nguồn ngắt Pin Change 8.

- **Port C (PC7:PC0)**

Port C là port I/O 8 bit 2 chiều có điện trở kéo lên bên trong, có thể lập trình riêng cho từng bit. Khi reset các chân port là 3 trạng thái.

Port C cũng thực hiện các chức năng khác như sau:

Bảng 2.4 Các chức năng khác của chân Port C

Chân port	Chức năng luân phiên
PC7	TOSC2 (Timer Oscillator pin 2) PCINT23 (Pin Change Interrupt 23)
PC6	TOSC1 (Timer Oscillator pin 1) PCINT22 (Pin Change Interrupt 22)
PC5	TDI (JTAG Test Data Input) PCINT21 (Pin Change Interrupt 21)
PC4	TDO (JTAG Test Data Output) PCINT20 (Pin Change Interrupt 20)
PC3	TMS (JTAG Test Mode Select) PCINT19 (Pin Change Interrupt 19)
PC2	TCK (JTAG Test Clock) PCINT18 (Pin Change Interrupt 18)
PC1	SDA (2-wire Serial Bus Data Input/Output Line) PCINT17 (Pin Change Interrupt 17)
PC0	SCL (2-wire Serial Bus Clock Line) PCINT16 (Pin Change Interrupt 16)

- **PC7: TOSC2/PCINT23**

TOSC2: chân 2 bộ dao động định thi.

PCINT23: nguồn ngắt Pin Change 23.

- **PC6: TOSC1/PCINT22**

TOSC1: chân 1 bộ dao động định thi.

PCINT22: nguồn ngắt Pin Change 22.

- **PC5: TDI/PCINT21**

**TDI:** ngõ vào dữ liệu kiểm tra JTAG.  
**PCINT21:** nguồn ngắt Pin Change 21.

- **PC4: TDI/PCINT20**

**TDO:** ngõ ra dữ liệu kiểm tra JTAG.  
**PCINT20:** nguồn ngắt Pin Change 20.

- **PC3: TMS/PCINT19**

**TMS:** chọn chế độ kiểm tra JTAG.  
**PCINT19:** nguồn ngắt Pin Change 19.

- **PC2: TCK/PCINT18**

**TCK:** xung nhịp kiểm tra JTAG.  
**PCINT18:** nguồn ngắt Pin Change 18.

- **PC1: SDA/PCINT17**

**SDA:** đường dữ liệu trong giao tiếp nối tiếp 2 dây (I2C).  
**PCINT17:** nguồn ngắt Pin Change 17.

- **PC0: SCL/PCINT16**

**SCL:** xung nhịp trong giao tiếp nối tiếp 2 dây (I2C).  
**PCINT16:** nguồn ngắt Pin Change 16.

- **Port D (PD7:PD0)**

Port D là port I/O 8 bit 2 chiều có điện trở kéo lên bên trong, có thể lập trình riêng cho từng bit. Khi reset các chân port là 3 trạng thái.

Port D cũng thực hiện các chức năng khác như sau:

**Bảng 2.5** Các chức năng khác của chân Port D

Chân port	Chức năng luân phiên
PD7	OC2A (Timer/Counter2 Output Compare Match A Output) PCINT31 (Pin Change Interrupt 31)
PD6	ICP1 (Timer/Counter1 Input Capture Trigger) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT30 (Pin Change Interrupt 30)
PD5	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT29 (Pin Change Interrupt 29)
PD4	OC1B (Timer/Counter1 Output Compare Match B Output) XCK1 (USART1 External Clock Input/Output) PCINT28 (Pin Change Interrupt 28)
PD3	INT1 (External Interrupt1 Input) TXD1 (USART1 Transmit Pin) PCINT27 (Pin Change Interrupt 27)
PD2	INT0 (External Interrupt0 Input) RXD1 (USART1 Receive Pin) PCINT26 (Pin Change Interrupt 26)
PD1	TXD0 (USART0 Transmit Pin) PCINT25 (Pin Change Interrupt 25)
PD0	RXD0 (USART0 Receive Pin) PCINT24 (Pin Change Interrupt 24)

- **PD7: OC2A/PCINT31**

**OC2A:** ngõ ra so sánh ngõ ra Match A của Bộ định thi/Bộ đếm 2. Chân này cũng là ngõ ra khi bộ định thi ở chế độ PWM.

**PCINT31:** nguồn ngắt Pin Change 31.

- **PD6: ICP1/OC2B/PCINT30**

**ICP1:** chân input capture của Bộ định thời/Bộ đếm 1.

**OC2B:** ngõ ra so sánh ngõ ra Match B của Bộ định thi/Bộ đếm 2. Chân này cũng là ngõ ra khi bộ định thi ở chế độ PWM..

**PCINT30:** nguồn ngắt Pin Change 30.

- **PD5: OC1A/PCINT29**

**OC1A:** ngõ ra so sánh ngõ ra Match A của Bộ định thi/Bộ đếm 1. Chân này cũng là ngõ ra khi bộ định thi ở chế độ PWM..

**PCINT29:** nguồn ngắt Pin Change 29.

- **PD4: OC1B/XCK1/PCINT28**

**OC1B:** ngõ ra so sánh ngõ ra Match B của Bộ định thi/Bộ đếm 1. Chân này cũng là ngõ ra khi bộ định thi ở chế độ PWM.

**XCK1:** xung nhịp bên ngoài cho USART 1 khi dùng ở chế độ đồng bộ.

**PCINT28:** nguồn ngắt Pin Change 28.

- **PD3: INT1/TXD1/PCINT27**

**INT1:** ngắt ngoài 1.

**TXD1:** ngõ ra phát dữ liệu của USART 1.

**PCINT27:** nguồn ngắt Pin Change 27.

- **PD2: INT0/RXD1/PCINT26**

**INT0:** ngắt ngoài 0.

**TXD1:** ngõ vào thu dữ liệu của USART 1.

**PCINT26:** nguồn ngắt Pin Change 26.

- **PD1: TXD0/PCINT25**

**TXD0:** ngõ ra phát dữ liệu của USART 0.

**PCINT25:** nguồn ngắt Pin Change 25.

- **PD1: RXD0/PCINT24**

**RXD0:** ngõ vào thu dữ liệu của USART 0.

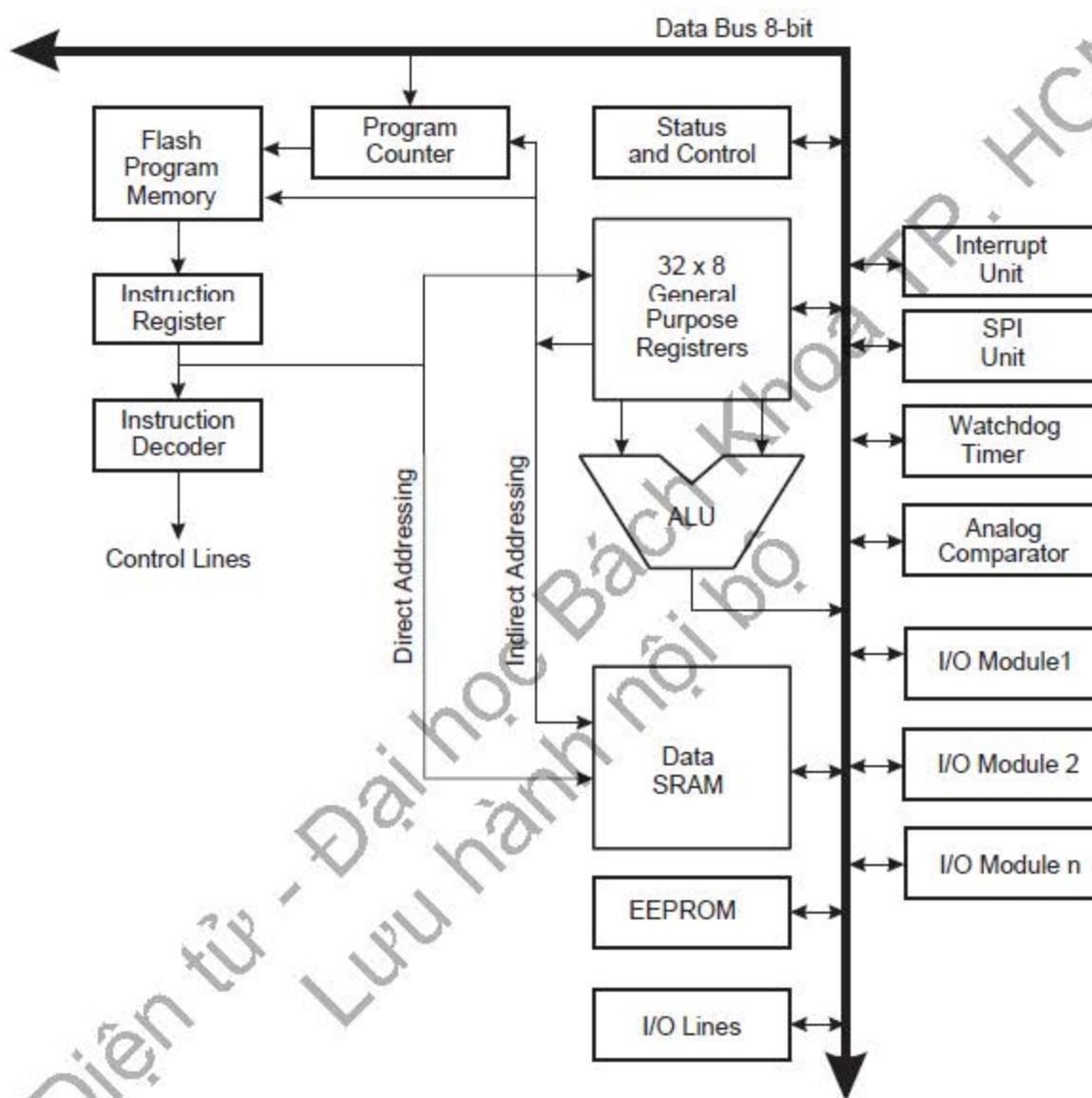
**PCINT24:** nguồn ngắt Pin Change 24.

### Câu hỏi ôn tập

1. Cho biết dung lượng các bộ nhớ trong của ATmega324P.
2. Liệt kê 3 thành phần ngoại vi có trong ATmega324P.
3. Chân port nào là ngõ vào của các bộ Timer?
4. Port nào là 8 kênh vào của bộ ADC?
5. Cho biết các phương thức giao tiếp nối tiếp của ATmega324P.
6. Chân port nào tương ứng với ngõ vào các ngắt ngoài INT0, INT1 và INT2?

### 2.3 LÕI CPU (CPU CORE)

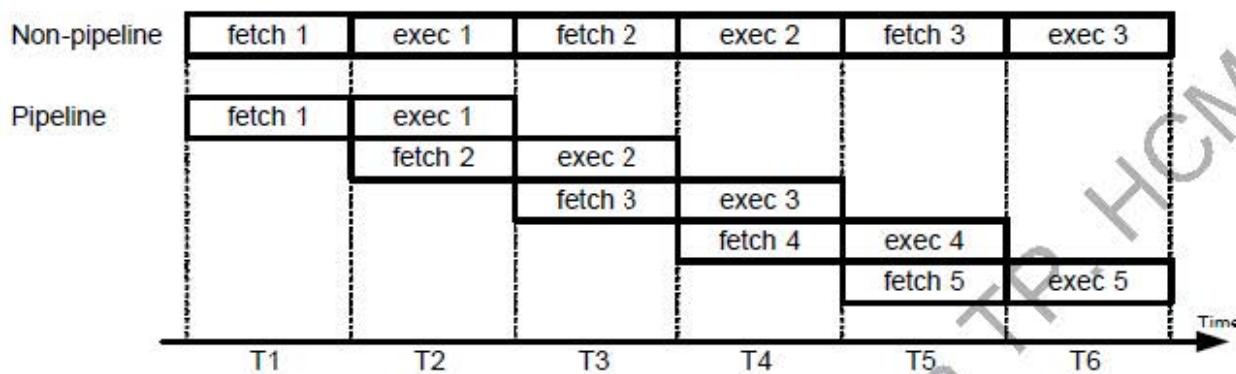
Chức năng chính của CPU là đảm bảo việc thực thi đúng chương trình. CPU phải có khả năng truy xuất các bộ nhớ, thực hiện các phép toán, điều khiển các ngoại vi và xử lý các ngắt.



Hình 2.3 Sơ đồ khái niệm về cấu trúc AVR

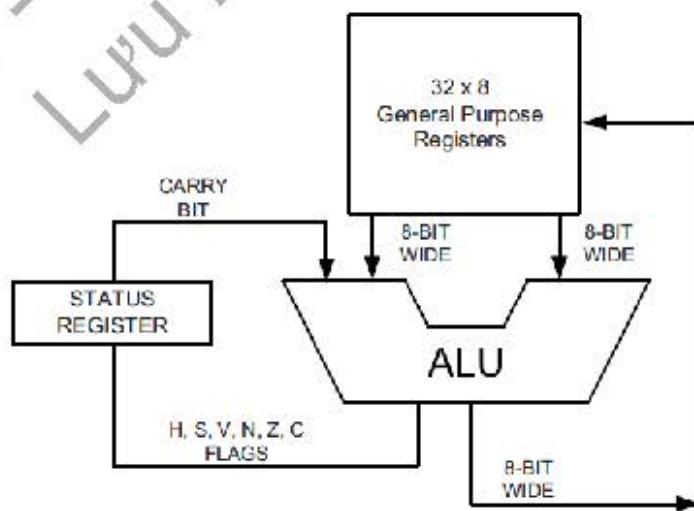
- Thành phần quan trọng nhất trong lõi AVR là thanh PC – Program Counter (bộ đếm chương trình). Bộ đếm chương trình được CPU sử dụng để trỏ đến địa chỉ của lệnh tiếp theo sẽ được thực thi. Khi CPU lấy mã lệnh từ ROM chương trình, bộ đếm chương trình sẽ tự động tăng 1 lên để trỏ đến lệnh tiếp theo. Trong vi điều khiển AVR, mỗi vị trí bộ nhớ Flash có chiều rộng là 2 byte. ATmega324P có Flash ROM là 32K byte. Flash được tổ chức là từng word 2 byte,  $32K \times 8 = 16K \times 16$ . Như vậy bộ đếm chương trình có 14 bit ( $2^{14} = 16K$  vị trí bộ nhớ). Khi reset hay cấp nguồn hoạt động, giá trị đầu của PC = 0x0000, do vậy lệnh đầu tiên thực hiện là lệnh chứa tại vị trí 0x0000 của Flash.
- Các lệnh trong bộ nhớ chương trình được thực thi với kỹ thuật đường ống đơn cấp. Trong khi 1 lệnh đang được thực thi (execute), thì lệnh kế tiếp được lấy mã (fetch) trước từ bộ nhớ chương trình. Khái niệm này cho phép các lệnh được thực thi trong

mỗi chu kỳ xung nhịp. Bộ nhớ chương trình là Flash Rom có thể tái lập trình ngay trong hệ thống mà không cần phải tháo chip ra khỏi hệ thống.



**Hình 2.4** Kỹ thuật đường ống (pipeline) và không đường ống (non-pipeline)

- Tập thanh ghi truy xuất nhanh gồm 32 thanh ghi đa dụng 8 bit (GPRs: General Purpose Registers) với thời gian truy xuất 1 chu kỳ xung. Việc này cho phép ALU thực hiện phép toán trên 2 toán hạng từ tập thanh ghi và kết quả cắt lại vào tập thanh ghi chỉ trong 1 chu kỳ xung nhịp. Có 6 thanh ghi trong tập thanh ghi có thể dùng như 3 thanh ghi 16 bit X, Y và Z (sẽ được mô tả trong phần sau).
- Bộ ALU hỗ trợ các phép toán số học và logic giữa các thanh ghi hay giữa thanh ghi và hằng số. Các phép toán của ALU được chia thành ba nhóm chính: phép toán số học, phép toán logic và xử lý bit. ALU cung cấp một bộ nhân mạnh hỗ trợ phép nhân có dấu /không dấu và định dạng phân số. Sau khi thực hiện phép toán số học, 1 thanh ghi trạng thái sẽ cập nhật thông tin kết quả của phép toán.



**Hình 2.5** ALU và tập thanh ghi

- Không gian bộ nhớ Flash chương trình chia thành 2 phần: chương trình khởi động (boot program) và chương trình ứng dụng (application program).

- Khi ngắt hay gọi chương trình con, nội dung thanh ghi PC được cất vào ngăn xếp (stack). Ngăn xếp được phân bổ hiệu quả trong vùng dữ liệu SRAM, do đó kích thước ngăn xếp chỉ bị giới hạn bởi kích thước SRAM và cách dùng SRAM. Tất cả các chương trình ứng dụng phải khởi tạo SP trong phần khởi động (trước khi các chương trình con hay ngắt được thực thi).
- Không gian bộ nhớ I/O gồm 64 + 160 địa chỉ cho các chức năng điều khiển CPU hay ngoại vi như các thanh ghi trạng thái, thanh ghi điều khiển SPI và các chức năng I/O khác.

## 2.4 BỘ NHỚ AVR

Bộ nhớ AVR có 2 không gian bộ nhớ là bộ nhớ dữ liệu và bộ nhớ chương trình.

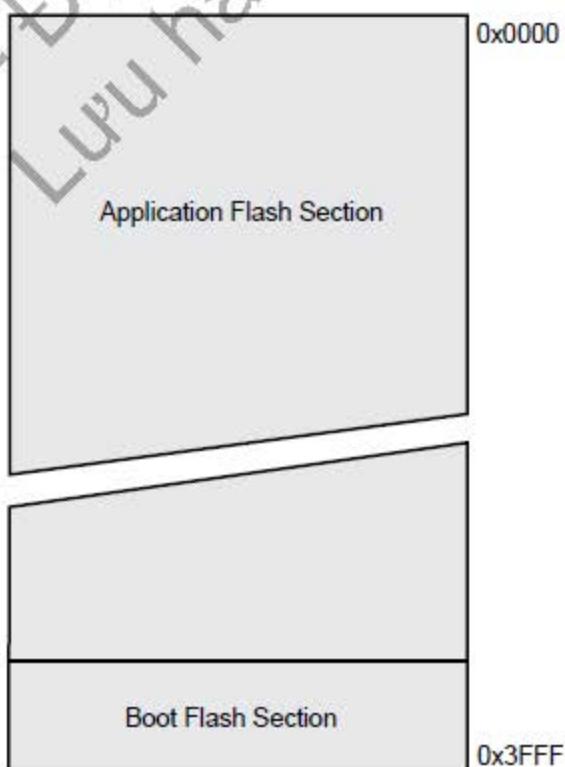
### 2.4.1 Bộ nhớ chương trình Flash

ATmega324P có 32KB bộ nhớ Flash để làm bộ nhớ chương trình. Tất cả lệnh AVR có độ dài là 16 bit hay 32 bit. Bộ nhớ Flash được tổ chức theo vị trí là 16 bit (2 Byte).

Bộ nhớ Flash 32KB = 16K x 2B, do vậy sẽ có  $2^{14}$  vị trí. Ta có địa chỉ từ 0x0000 – 0x3FFF.

Bộ nhớ chương trình Flash chứa 2 phần: ứng dụng (Application) và khởi động (Boot). Kích thước mỗi vùng có thể tùy chọn.

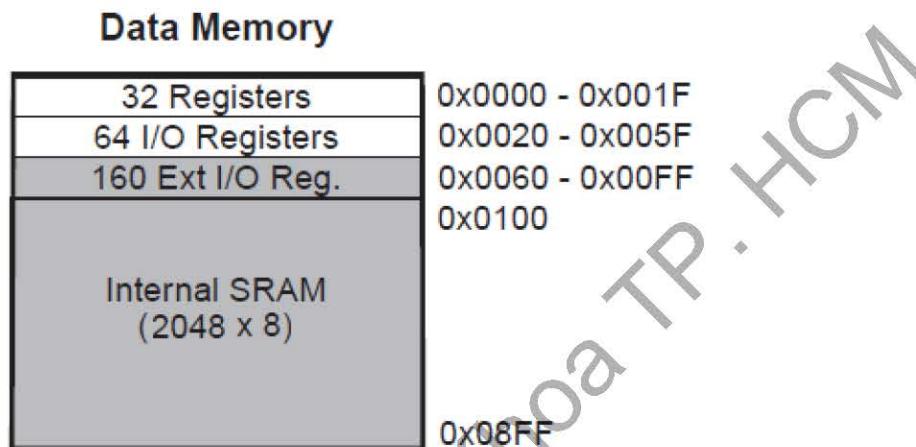
Bộ nhớ Flash có thể chịu được 10,000 chu kỳ ghi/xóa. Việc nạp chương trình vào bộ nhớ Flash có thể được lập trình song song hoặc nối tiếp qua giao tiếp JTAG hay SPI.



Hình 2.6 Bản đồ bộ nhớ chương trình

#### 2.4.2 Bộ nhớ dữ liệu SRAM

Bộ nhớ dữ liệu gồm 32 thanh ghi làm việc đa dụng, 64 thanh ghi I/O, 160 thanh ghi I/O mở rộng và 2KB SRAM nội. Địa chỉ bộ nhớ SRAM của ATmega324P từ 0x000 đến 0x08FF.



Hình 2.7 Bản đồ bộ nhớ dữ liệu của ATmeg324P

##### 2.4.2.1 Tập thanh ghi đa dụng (GPRs:General Purpose Registers)

CPU sử dụng các thanh ghi 8 bit này để chứa dữ liệu tạm thời dùng trong các phép toán số học và logic. Các thanh ghi này là R0 đến R31 có địa chỉ bộ nhớ tương ứng 0x00 đến 0x1F cho tất cả họ AVR. Tập thanh ghi được chia thành 2 phần, mỗi phần có 16 thanh ghi: R0 ÷ R15 và R16 ÷ R31.

General Purpose Working Registers	7	0	Addr.	
	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Hình 2.8 Tập thanh ghi đa dụng(GPRs)

Trong tập lệnh AVR quy định có 1 số lệnh chỉ được sử dụng thanh ghi R16÷R31 như LDI, IN, OUT...

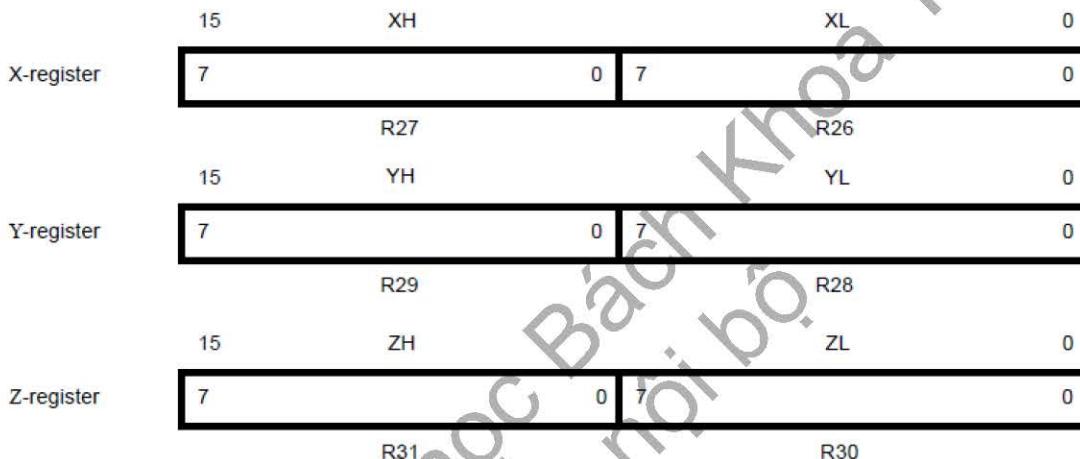
##### Ví dụ 2.2 Các lệnh sử dụng thanh ghi đa dụng

**LDI R16, 0x25** ; nạp giá trị 25H vào thanh ghi R16

**LDI R17, 125** ; nạp giá trị 125 vào thanh ghi R17  
**ADD R10, R16** ; cộng giá trị trong thanh ghi R10 và R16,  
 kết quả chừa lại vào thanh ghi R10  
**INC R4** ;tăng nội dung R4 thêm 1  
**OUT PORTB,R20** ;xuất nội dung R20 ra PortB

Có 1 số thanh ghi có chức năng bổ sung riêng.Ví dụ như kết quả phép nhân 2 số nhị phân 8 bit là số nhị phân 16 bit mặc định trả về R1:R0 (R1 byte cao, R0 byte thấp)

Có 6 thanh ghi R26 đến R31 được ghép thành 3 thanh ghi 16 bit là thanh ghi X, Y và Z. Các thanh ghi này là con trỏ thanh ghi địa chỉ gián tiếp dùng để truy xuất bộ nhớ dữ liệu.



Hình 2.9 Các thanh ghi X, Y, Z

#### 2.4.2.2 Bộ nhớ I/O (I/O Memory)

Bộ nhớ I/O là các thanh ghi 8 bit có địa chỉ bộ nhớ từ 0x20 đến 0xFF. Các thanh ghi I/O này có chức năng riêng để điều khiển vi điều khiển hay ngoại vi. Ta sẽ xem xét chức năng cụ thể của 1 số thanh ghi trong mục 2.5.

Bộ nhớ I/O chia làm 2 phần: thanh ghi I/O chuẩn (Standard I/O registers) và thanh ghi I/O mở rộng.

- Thanh ghi I/O chuẩn: gồm 64 thanh ghi có địa chỉ bộ nhớ từ 0x20 – 0x5F. Các thanh ghi này ngoài địa chỉ bộ nhớ còn có địa chỉ I/O từ 0x00 – 0x3F (chênh lệch 0x20 so với địa chỉ bộ nhớ tương ứng). Địa chỉ này dùng khi sử dụng các lệnh IN, OUT để làm việc với thanh ghi. Ngoài ra các thanh ghi I/O có địa chỉ I/O từ 0x00 – 0x1F có thể truy xuất bit trực tiếp trong các lệnh xử lý bit như CBI, SBI, ..
- Thanh ghi I/O mở rộng: gồm 160 thanh ghi có địa chỉ bộ nhớ từ 0x60 – 0xFF. Để truy xuất các thanh ghi I/O mở rộng phải dùng các lệnh STS/LD như truy xuất trực tiếp địa chỉ trực tiếp SRAM.

Nói chung, các thanh ghi I/O có thể trao đổi dữ liệu với tập thanh ghi làm việc đa dụng bằng các lệnh LD/LDS/LDD và ST/STS/STD. Khi đó dùng địa chỉ bộ nhớ khi truy xuất thanh ghi I/O. Trong chương 3 sẽ trình bày chi tiết vấn đề này.

Trong vùng bộ nhớ I/O còn có một số địa chỉ dành riêng (reserved) để tương thích với các thiết bị phát triển trong tương lai.

Địa chỉ		Tên Thanh ghi
Bộ nhớ	I/O	
0x20	0x00	PINA
0x21	0x01	DDRA
0x22	0x02	PORTA
0x23	0x03	PINB
0x24	0x04	DDRB
0x25	0x05	PORTB
0x26	0x06	PINC
0x27	0x07	DDRC
0x28	0x08	PORTC
0x29	0x09	PIND
0x2A	0x0A	DDRD
0x2B	0x0B	PORTD
0x2C		Reserved
0x34		
0x35	0x15	TIFR0
0x36	0x16	TIFR1
0x37	0x17	TIFR2
0x38		Reserved
0x3A		
0x3B	0x1B	PCIFR
0x3C	0x1C	EIFR
0x3D	0x1D	EIMSK
0x3E	0x1E	GPIO0
0x3F	0x1F	EECR
0x40	0x20	EEDR
0x41	0x21	EEARL và EEARH
0x42	0x22	
0x43	0x23	GTCCR
0x44	0x24	TCCR0A
0x45	0x25	TCCR0B
0x46	0x26	TCNT0
0x47	0x27	OCR0A
0x48	0x28	OCR0B
0x49		Reserved
0x4A	0x2A	GPIO1
0x4B	0x2B	GPIO2
0x4C	0X2C	SPCR0
0x4D	0x2D	SPSR0
0x4E	0x2E	SPDR0
0x4F	0x2F	Reserved
0x50	0x30	ACSR
0x51	0x31	OCDR
0x52	0x32	Reserved
0x53	0x33	SMCR

Địa chỉ		Tên Thanh ghi
Bộ nhớ	I/O	
0x54	0x34	MCUSR
0x55	0x35	MCUCR
0x56		Reserved
0x57	0x37	SPMCSR
0x58		Reserved
0x5C		
0x5D	0x3D	SPL và SPH
0x5E	0x3E	
0x5F	0x3F	SREG
0x60		WDTCSR
0x61		CLKPR
0x62		Reserved
0x63		Reserved
0x64		PRR0
0x65		Reserved
0x66		OSCCAL
0x67		Reserved
0x68		PCICR
0x69		EICRA
0x6A		Reserved
0x6B		PCMSK0
0x6C		PCMSK1
0x6D		PCMSK2
0x6E		TIMSK0
0x6F		TIMSK1
0x70		TIMSK2
0x71		Reserved
0x72		Reserved
0x73		PCMSK3
0x74		Reserved
0x77		
0x78		ADCL và ADCH
0x79		
0x7A		ADCSRA
0x7B		ADCSRB
0x7C		ADMUX
0x7D		Reserved
0x7E		DIDR0
0x7F		DIDR1
0x80		TCCR1A
0x81		TCCR1B
0x82		TCCR1C
0x83		Reserved

Địa chỉ		Tên Thanh ghi
Bộ nhớ	I/O	
0x84		TCNT1L và TCNT1H
0x85		
0x86		ICR1L và ICR1H
0x87		
0x88		OCR1AL và OCR1AH
0x89		
0x8A		OCR1BL và OCR1BH
0x8B		
0x8C		Reserved
0xAF		
0xB0		TCCR2A
0xB1		TCCR2B
0xB2		TCNT2
0xB3		OCR2A
0xB4		OCR2B
0xB5		Reserved
0xB6		ASSR
0xB7		Reserved
0xB8		TWBR
0xB9		TWSR
0xBA		TWAR
0xBB		TWDR
0xBC		TWCR
0xBD		TWAMR
0xBE		Reserved
0xBF		Reserved
0xC0		UCSR0A
0xC1		UCSR0B
0xC2		UCSR0C
0xC3		Reserved
0xC4		UBRR0L và UBRR0H
0xC5		
0xC6		UDR0
0xC7		Reserved
0xC8		UCSR1A
0xC9		UCSR1B
0xCA		UCSR1C
0xCB		Reserved
0xCC		UBRR1L và UBRR1H
0xCD		
0xCE		UDR1
0xCF		Reserved
...		
0xFF		

Hình 2.10 Các thanh ghi I/O

Xem thêm phụ lục tóm tắt các thanh ghi trong bộ nhớ I/O của AVR ATmega324P.

#### 2.4.2.3 SRAM nội

ATmega324P có 2KB Ram nội có địa chỉ từ 0x0100 – 0x08FF. Vùng nhớ này có thể được truy xuất trực tiếp hoặc gián tiếp. Bộ nhớ này cũng được dùng làm ngăn xếp khi dùng chương trình con hay chương trình ngắn.

#### 2.4.3 Bộ nhớ dữ liệu EEPROM

ATmega324P có 1KB bộ nhớ dữ liệu EEPROM có địa chỉ từ 0x0000 - 0x03FF. EEPROM chịu được ít nhất 100,000 chu kỳ ghi/xóa. Bộ nhớ EEPROM được sử dụng để lưu trữ dữ liệu ít khi bị thay đổi và không bị mất khi tắt nguồn.

Để truy xuất vùng nhớ này, sử dụng các thanh ghi chức năng trong vùng bộ nhớ I/O (trình bày trong mục 2.5.4).

#### Câu hỏi ôn tập

1. Bộ nhớ nào của ATmega324P có tổ chức là 8 bit? 16 bit?
2. Bộ nhớ SRAM có tổng cộng bao nhiêu Byte? (nếu tính luôn tập thanh ghi và thanh ghi I/O)
3. Tập thanh ghi làm việc đa dụng GPR có địa chỉ bộ nhớ là bao nhiêu?
4. Thanh ghi R23 có địa chỉ bộ nhớ là bao nhiêu?
5. Các thanh ghi 16 bit X, Y, Z được ghép từ các thanh ghi nào?
6. Làm cách nào để tìm địa chỉ bộ nhớ từ địa chỉ I/O của 1 thanh ghi I/O.
7. Các lệnh IN/OUT chỉ được dùng trên các thanh ghi I/O nào? (nếu địa chỉ bộ nhớ, địa chỉ I/O)
8. Các thanh ghi I/O nào có truy xuất bit? (nếu địa chỉ bộ nhớ, địa chỉ I/O)?
9. Nếu tên 3 thanh ghi I/O có địa chỉ I/O và 3 thanh ghi không có địa chỉ I/O.
10. Địa chỉ cuối của vùng SRAM là bao nhiêu?

### 2.5 CHỨC NĂNG THANH GHI I/O

Phần này sẽ khảo sát một số thanh ghi I/O thông thường. Các thanh ghi khác sẽ được đề cập trong các chương tiếp theo.

#### 2.5.1 Thanh ghi trạng thái – SREG (Status Register)

Thanh ghi trạng thái chứa thông tin về kết quả của lệnh số học/logic hoặc điều kiện được thực hiện gần nhất. Thông tin này chỉ báo trạng thái kết quả hoặc được dùng làm thay đổi hướng chạy chương trình để thực hiện các phép toán điều kiện. Các bit trong thanh ghi SREG thường được gọi là các cờ báo trạng thái. Việc cập nhật các cờ này sau từng lệnh sẽ được trình bày chi tiết trong chương 3.

Bit	7	6	5	4	3	2	1	0	SREG
0x3F (0x5F)	I	T	H	S	V	N	Z	C	
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

Hình 2.11: Tổ chức thanh ghi SREG

- Bit 7 – I : Cho phép ngắt toàn cục (Global Interrupt Enable)

Bit I bằng 1 để cho phép chương trình có sử dụng các ngắt và bằng 0 thì cấm sử dụng các ngắt. Mỗi nguồn ngắt lại được điều khiển riêng (đề cập chi tiết trong chương 10). Bit I sẽ bị xóa về 0 bằng phần cứng khi chương trình ngắt thực hiện và sẽ đặt lên 1 khi thực hiện lệnh RETI để cho phép các ngắt tiếp theo. Bit I cũng được đặt lên 1 bằng lệnh SEI và xóa về 0 bằng lệnh CLI.

- Bit 6 – T : bit sao chép (Copy Storage)

Các lệnh sao chép bit sử dụng bit T để chứa giá trị. Lệnh BLD (Bit Load) cho phép sao chép bit T vào 1 bit bất kỳ trong 1 thanh ghi thuộc tập thanh ghi GPRs. Lệnh BST (Bit Store) cho phép sao chép 1 bit bất kỳ trong 1 thanh ghi thuộc tập thanh ghi GPRS vào bit T.

- Bit 5 – H : Cờ nhớ phân nửa (Half Carry Flag)

Cờ H là cờ nhớ phân nửa trong các phép toán số học. Cờ H=1 khi có tràn từ bit 3 sang bit 4. Thường dùng khi dùng phép toán số học trên số BCD.

- Bit 4 – S : Cờ dấu (Sign Flag)

Cờ S báo dấu chính xác của kết quả, S=0 báo kết quả dương, S=1 báo kết quả âm.

Có thể xem  $S = N \oplus V$  (xem phần sau)

- Bit 3 – V : Cờ tràn bù 2 (Two's Complement Overflow Flag)

Cờ V=1 khi kết quả của 1 phép toán số có dấu lớn hơn phạm vi biểu diễn của nó gọi là tràn số có dấu dạng bù 2. Hay nói cách khác, cờ V=1 khi một trong hai trường hợp sau xảy ra (xem cờ C phần sau):

- Kết quả có tràn từ bit 6 qua bit 7, nhưng không có tràn từ bit 7 (C=0)

- Kết quả có tràn từ bit 7 (C=1), nhưng không có tràn từ bit 6 qua bit 7

Như vậy nếu kết quả đồng thời có tràn từ bit 6 qua bit 7 và tràn từ bit 7 (C=1) thì cờ V=0.

- Bit 2 – N : Cờ âm (Negative Flag)

Cờ N cho biết kết quả âm trong phép toán số học. Cờ N có giá trị như bit 7 của kết quả.

Cờ N = 0 thì kết quả là số dương, N = 1 thì kết quả là số âm.

- Bit 1 – Z : Cờ không (Zero Flag)

Cờ Z cho biết kết quả bằng 0 trong các phép toán số học và logic. Kết quả bằng 0 thì cờ Z = 1, ngược lại kết quả khác 0 thì Z = 0.

- Bit 0 – C : Cờ nhớ (Carry Flag)

Cờ C là bit nhớ từ bit 7 khi thực hiện phép toán cộng hay trừ. Cờ C=1 khi kết quả tràn từ bit 7 sang.

**Ví dụ 2.3** Xác định nội dung của SREG sau khi thực hiện xong đoạn chương trình

LDI R20, 0x54

LDI R25, 0xC4

ADD R20, R25

Vị trí bit	7	6	5	4	3	2	1	0	Số có dấu
Bit nhớ	1	1	0	0	0	1	0	0	+84
R20	0	1	0	1	0	1	0	0	-60
R25	1	1	0	0	0	1	0	0	
<b>R20 ← R20+R25</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>+24</b>

Thực hiện phép cộng 2 số, ta có cờ C = 1, cờ H = 0, và cờ N = 0.

Xét số có dấu thì kết quả không bị tràn: cờ V = 0.

Do đó cờ S = 0 ( $S = N \oplus V$ )

SREG	I	T	H	S	V	N	Z	C
	-	-	0	0	0	0	0	1

### 2.5.2 Con trỏ ngăn xếp – SP (Stack Pointer)

Ngăn xếp là 1 phần bộ nhớ thường là RAM, được sử dụng để chứa dữ liệu tạm thời. Ngăn xếp thường được dùng khi có chương trình ngắn và khi gọi chương trình con. Ngoài ra còn được sử dụng khi dùng lệnh PUSH và POP.

Thanh ghi SP luôn trỏ đến đỉnh ngăn xếp (T.O.S – Top Of Stack). Nội dung thanh ghi SP chứa địa chỉ của ô nhớ đang là đỉnh ngăn xếp. Khi có lệnh PUSH dữ liệu sẽ được đẩy vào ngăn xếp lúc đó đỉnh ngăn xếp tăng; với AVR thì đồng nghĩa với việc làm giảm nội dung con trỏ SP. Khi có lệnh POP thì dữ liệu được lấy ra khỏi ngăn xếp và đỉnh ngăn xếp giảm xuống; đồng nghĩa với việc tăng nội dung con trỏ SP. Lưu ý rằng ở AVR ngăn xếp được thực hiện từ địa chỉ cao đến địa chỉ thấp hơn.

Ngăn xếp có cấu trúc LIFO – Last In First Out. Dữ liệu đưa vào sau cùng sẽ nằm đỉnh ngăn xếp, do đó khi được lấy ra thì dữ liệu đó sẽ được lấy ra trước.

Ở AVR, thanh ghi SP là 2 thanh ghi 8 bit SPH (Stack Pointer High) và SPL (Stack Pointer Low) kết hợp lại. SPH và SPL thuộc vùng thanh ghi I/O.

Bit	15	14	13	12	11	10	9	8	SPH
0x3E (0x5E)	-	-	-		SP11	SP10	SP9	SP8	
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	1	0	0	0	
	1	1	1	1	1	1	1	1	

Hình 2.12: Tổ chức thanh ghi SP

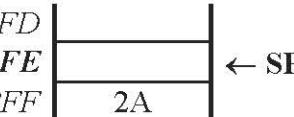
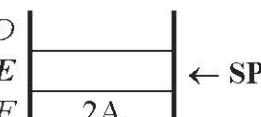
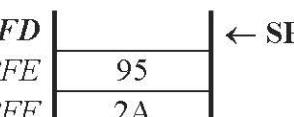
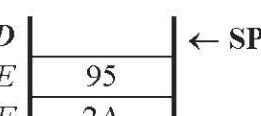
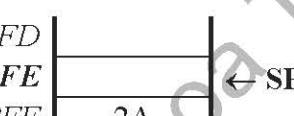
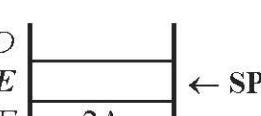
ATmega324P có ngăn xếp trong vùng SRAM 2KB, có địa chỉ từ 0x0100 đến 0x08FF. Do đó thanh ghi SP chỉ dùng 12 bit: SP11.. SP0 để chứa địa chỉ đỉnh ngăn xếp.

Chú ý là ngăn xếp của AVR khi cất dữ liệu vào (lệnh PUSH) ô nhớ địa chỉ trỏ bởi nội dung SP xong, nội dung SP sẽ giảm 1, khi lấy dữ liệu từ ngăn xếp (lệnh POP), nội dung SP tăng 1 rồi mới lấy dữ liệu từ ô nhớ địa chỉ trỏ bởi SP.

Vùng ngăn xếp phải được xác định trong chương trình trước khi gọi chương trình con hay ngắn được cho phép. Với ATmega324P sau khi reset, giá trị đầu của thanh ghi SP là 0x08FF, bằng địa chỉ cao nhất của vùng SRAM, nên không cần khởi động lại vùng ngăn xếp.

**Ví dụ 2.4** Với nội dung các thanh ghi SP = 0x08FF, R20 = 0x2A, R21 = 0x95, R22 = 0x4C. Cho biết nội dung ngăn xếp và các thanh ghi sau khi thực hiện tuần tự các lệnh.

Ngăn xếp	LỆNH	Ngăn xếp	Thanh ghi
----------	------	----------	-----------

trước khi thực hiện lệnh		sau khi thực hiện lệnh	
	PUSH R20		R20 = \$2A R21 = \$95 R22 = \$4C
	PUSH R21		R20 = \$2A R21 = \$95 R22 = \$4C
	POP R20		R20 = \$95 R21 = \$95 R22 = \$4C
	POP R22		R20 = \$95 R21 = \$95 <b>R22 = \$2A</b>

- **Chú ý:** trong họ AVR có một số MCU sau khi reset ghi SP = 0x0000. Do vậy khi sử dụng vùng ngăn xếp để thực hiện chương trình con hay ngắn, cần chuyển SP tới địa chỉ cao nhất của vùng SRAM. Tổng quát, để có thể dùng chung cho các loại AVR có dung lượng SRAM khác nhau, sử dụng đoạn chương trình sau:

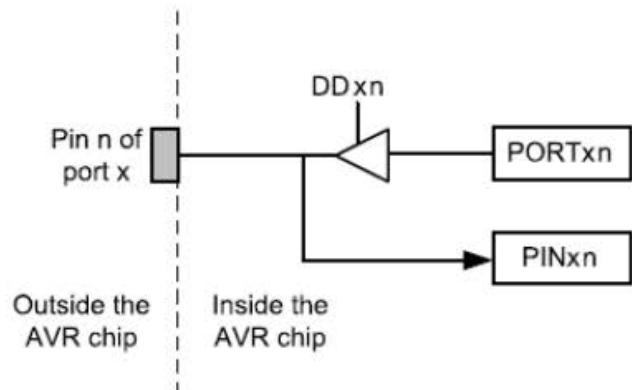
```
.ORG 0 ; chương trình bắt đầu từ địa chỉ 0x0000
LDI R16, HIGH(RAMEND) ; nạp SPH giá trị byte cao của RAMEND
OUT SPH, R16
LDI R16, LOW(RAMEND) ; nạp SPL giá trị byte thấp của RAMEND
OUT SPL, R16
```

Ký hiệu RAMEND là địa chỉ cuối của vùng SRAM tương đương với mỗi AVR.

### 2.5.3 Các thanh ghi cổng xuất nhập (I/O Port) - DDRx, PORTx, PINx

ATmega324P có 4 port: PortA, PortB, PortC và PortD. Các chân port này có nhiều chức năng riêng để làm việc với ngoại vi. Nếu không dùng các chức năng này thì các chân port có thể dùng làm chân xuất/nhập số.

Để sử dụng các port này, thì cần phải lập trình chức năng cho mỗi chân port. Mỗi port có 3 thanh ghi liên quan là DDRx, PORTx, PINx (với x là A, B, C, D). Chú ý là mỗi chân của mỗi port có thể điều khiển xuất/nhập riêng lẻ.



**Hình 2.13:** Cấu trúc cơ bản chân port xuất nhập (The I/O Port)

Ký hiệu các bit  $x\ n$  với  $x$  là tên các port (A, B, C, D) và  $n$  là số thứ tự chân trong port ( $0, 1, \dots, 7$ )

#### 2.5.3.1 Thanh ghi hướng dữ liệu – DDRx (Data Direction Register)

ATmega324P có 4 Port, mỗi Port có 8 đường I/O là tín hiệu 2 chiều. Tuy nhiên tại mỗi thời điểm các đường này chỉ nhập (input) hay xuất (output). Do vậy trước khi dùng chúng cần phải chọn cấu hình là nhập hay xuất bằng thanh ghi DDRx.

Bit	7	6	5	4	3	2	1	0	
0x01 (0x21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

**Hình 2.14:** Tổ chức các thanh ghi hướng dữ liệu I/O

Ghi các giá trị 1 vào thanh ghi DDRx tại vị trí nào thì bit tương ứng của chân port đó sẽ là chân xuất (output), ngược lại ghi giá trị 0 thì chân port sẽ là chân nhập (input).

Giá trị ban đầu của các thanh ghi này là 0x00 nên các port đều sẵn là chân nhập (input)

**Ví dụ 2.5:** Chọn chức năng xuất cho Port A và nhập cho Port C

```

LDI R16, 0xFF ; R16 = 0xFF = 0b11111111
OUT DDRA, R16 ; Port A được chọn là xuất
LDI R16, 0x00 ; R16 = 0x00 = 0b00000000
OUT DDRC, R16 ; Port C được chọn là nhập

```

### 2.5.3.2 Thanh ghi dữ liệu Port – PORTx (Data Register)

Khi các chân port được chọn là xuất (output), việc ghi 1 giá trị vào thanh ghi PORTx thì giá trị đó sẽ được xuất ra chân port tương ứng.

Bit	7	6	5	4	3	2	1	0	
0x02 (0x22)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORATA
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

Hình 2.15: Tổ chức các thanh ghi Port I/O

**Ví dụ 2.6:** Xuất giá trị \$55 ra port B

```

LDI R16, 0xFF ; R16 = 0xFF = 0b11111111
OUT DDRB, R16 ; Port B được chọn là xuất
LDI R16, 0x55 ; R16 = 0x55 = 0b01010101
OUT PORTB, R16 ; giá trị $55 được xuất ra Port B

```

### 2.5.3.2 Thanh ghi các chân nhập Port – PINx (Port Input Pins Register)

Khi các chân port được chọn là nhập (input), thì việc đọc nội dung thanh ghi PINx sẽ có được giá trị nhập về từ các chân port tương ứng.

Bit	7	6	5	4	3	2	1	0	
0x00 (0x20)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/W								
Initial Value	N/A								
Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W								
Initial Value	N/A								
Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R/W								
Initial Value	N/A								

Bit	7	6	5	4	3	2	1	0	PIND
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	
Read/Write	R/W								
Initial Value	N/A								

Hình 2.16: Tổ chức các thanh ghi Pin I/O

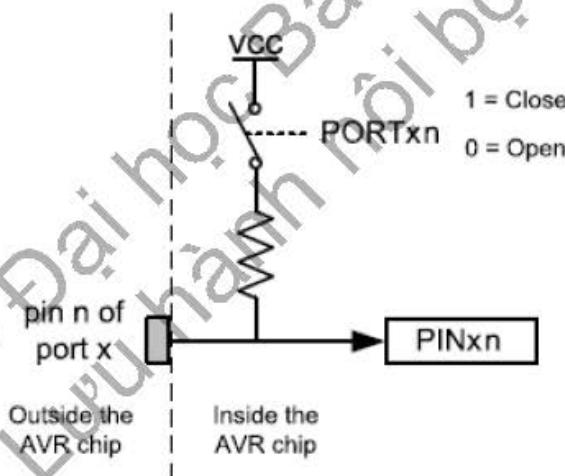
Ví dụ 2.7: Nhập dữ liệu 8 bit từ port C và xuất ra port D

```

LDI R16, 0x00 ; R16 = 0xFF = 0b00000000
OUT DDRC, R16 ; Port C được chọn là nhập
LDI R16, 0xFF ; R16 = 0xFF = 0b11111111
OUT DDRD, R16 ; Port D được chọn là xuất
IN R16, PINC ; nhập dữ liệu từ Port C và cất vào thanh ghi R16
OUT PORTD, R16 ; xuất giá trị ra Port D

```

Các chân port của AVR đều có điện trở kéo lên. Việc sử dụng điện trở kéo lên đảm bảo các ngõ vào ổn định tại mức logic mong đợi nếu các thiết bị bên ngoài không có kết nối hay ở trạng thái trơ kháng cao. Để sử dụng các điện trở này thì ghi các giá trị 1 vào thanh ghi Portx tương ứng. Nếu ghi giá trị 0 thì không cho phép sử dụng các điện trở kéo lên.



Hình 2.17: Điện trở kéo lên (The Pull-up Resistor)

Ví dụ 2.9: Tương tự như ví dụ 2.7, nhưng có cho phép Port C có điện trở kéo lên

```

LDI R16, 0x00 ; R16 = 0xFF = 0b00000000
OUT DDRC, R16 ; Port C được chọn là nhập
LDI R16, 0xFF ; R16 = 0xFF = 0b11111111
OUT PORTC, R16 ; cho sử dụng điện trở kéo lên ở Port C
OUT DDRD, R16 ; Port D được chọn là xuất
IN R16, PINC ; nhập dữ liệu từ Port C và cất vào thanh ghi R16
OUT PORTD, R16 ; xuất giá trị ra Port D

```

Chú ý là trong thanh ghi MCUCR (MCU Control Register) có bit PUD (Pull-up Disable) có thể cấm sử dụng điện trở kéo lên cho tất cả các chân port. Khi ghi giá trị 1 vào bit PUD, thì điện trở kéo lên bị cấm dù cho dù có DDxn = 0 và PORTxn = 1.

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	JTD	BODS	BODSE	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**Hình 2.18:** Thanh ghi MCUCR có bit PUD cho phép/cấm điện trở kéo lên Port Bảng 2.6 tóm tắt các tín hiệu điều khiển ở chân port.

**Bảng 2.6:** Các cấu hình chân port.

DDxn	PORTxn	PUD (MCUCR)	I/O	Điện trở Pull-up	Chân port
0	0	x	Nhập (Input)	Không	3 trạng thái Tri-state (Hi-Z)
0	1	0	Nhập (Input)	Có	Pxn sẽ kéo lên nguồn nếu bên ngoài kéo xuống thấp
0	1	1	Nhập (Input)	Không	3 trạng thái Tri-state (Hi-Z)
1	0	x	Xuất (Output)	Không	Ngõ ra mức thấp (Output Low)
1	1	x	Xuất (Output)	Không	Ngõ ra mức cao (Output High)

#### 2.5.4 Các thanh ghi EEPROM

ATmega324P có 1KB EEPROM. CPU dùng 3 thanh ghi để truy xuất vùng nhớ này:

Bit	15	14	13	12	11	10	9	8	
0x22 (0x42)	-	-	-	-	-	-	EEAR9	EEAR8	EEARH
0x21 (0x41)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R	R	R	R	R	R	R/W	R/W	
	R/W								
Initial Value	0	0	0	0	0	0	X	X	
	X	X	X	X	X	X	X	X	

(a) Thanh ghi EEARH\_EEARL

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	MSB							LSB	EEDR
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

(b) Thanh ghi EEDR

Bit	7	6	5	4	3	2	1	0	
0x1F (0x3F)	-	-	EEP1	EEP0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	X	X	0	0	X	0	

(c) Thanh ghi EECR

## Hình 2.19: Tổ chức các thanh ghi truy xuất EEPROM

### 2.5.4.1 Thanh ghi địa chỉ EEPROM – EEARH\_EEARL (EEPROM Address Register)

Cặp thanh ghi EEARH\_EEARL là thanh ghi EEAR 16 bit dùng để chứa địa chỉ của vùng nhớ cần truy xuất. Do EEPROM của ATmega324P chỉ có 1KB, vì vậy chỉ dùng 10 bit địa chỉ. Trước khi ghi/đọc EEPROM thì cần ghi địa chỉ của ô nhớ vào thanh ghi EEAR.

### 2.5.4.2 Thanh ghi dữ liệu EEPROM – EEDR (EEPROM Data Register)

Để ghi EEPROM thì dữ liệu được ghi vào thanh ghi EEDR. Để đọc EEPROM thì đọc thanh ghi EEDR.

### 2.5.4.3 Thanh ghi điều khiển EEPROM – EECR (EEPROM Control Register)

- Bit 5,4 – EEPM1:EPM0 : Các bit chọn kiểu lập trình (EEPROM Program Mode)  
Việc lập trình là 1 hoạt động gồm việc xóa và ghi bộ nhớ. Tuy nhiên có thể tách thành 2 hoạt động riêng lẻ. Các bit EEPM dùng để chọn hoạt động như bảng 2.7 với Fosc=8Mhz.

Bảng 2.7: Các bit chọn chế độ lập trình

EEPM[1:0]	Thời gian lập trình (Programming Time)	Hoạt động
00	3.4ms	Xóa và ghi trong cùng 1 hoạt động
01	1.8ms	Chỉ xóa
10	1.8ms	Chỉ ghi
11	-	Dự trữ cho ứng dụng tương lai

- Bit 3 – EERIE : Cho phép ngắt EEPROM (EEPROM Ready Interrupt Enable)  
Ghi 1 vào bit EERIE sẽ cho phép ngắt EEPROM hoạt động. Ghi 0 là cấm sử dụng ngắt EEPROM (xem chương 10).
- Bit 2 – EEMPE : Cho phép ghi chủ EEPROM (EEPROM Master Write Enable)  
Khi bit EEMPE được đặt lên 1, thì việc đặt bit EEPE lên 1 trong vòng bốn chu kỳ xung nhịp tiếp theo sẽ thực hiện việc ghi dữ liệu vào EEPROM ở vị trí đã chọn. Nếu EEMPE là 0, thì việc đặt EEPE không có tác dụng. Khi ghi 1 vào EEMPE, phần cứng sẽ xóa bit này về 0 sau bốn chu kỳ xung nhịp.
- Bit 1 – EEPE : Cho phép ghi EEPROM (EEPROM Write Enable)  
Bit EEPE là tín hiệu điều khiển việc ghi EEPROM. Khi địa chỉ và dữ liệu được thiết lập đúng, bit EEPE ghi thành 1 để cho phép ghi giá trị vào EEPROM. Bit EEMPE phải được ghi 1 trước khi EEPE được ghi 1; nếu không, không có quá trình ghi EEPROM nào diễn ra. Phần cứng sẽ xóa bit EEPE sau khi xong dữ liệu vào ô nhớ đã xác định. Khi đặt EEPE=1, CPU bị treo 2 chu kỳ xung nhịp trước khi thực hiện lệnh kế tiếp.

Để ghi EEPROM, cần theo các bước sau:

- Chờ đến khi bit EEPE=0.
- Ghi địa chỉ EEPROM vào thanh ghi EEAR.
- Ghi dữ liệu vào thanh ghi EEDR.
- Đặt bit EEMPE= 1.
- Trong vòng 4 chu kỳ xung nhịp sau khi đặt bit EEMPE=1 thì đặt bit EEPE =1.

Chú ý là ở bước 5, phải đặt bit EEPE=1 trong vòng 4 chu kỳ xung nhịp sau bước 4. Bởi vì sau thời gian đó phần cứng xóa bit EEMPE =0, việc đặt bit EEPE =1 cũng không thể ghi được EEPROM.

- Bit 0 – EERE : Cho phép đọc EEPROM (EEPROM Read Enable)

Bit EERE là tín hiệu điều khiển việc đọc EEPROM. Khi địa chỉ được thiết lập, việc đặt bit EERE =1 trong khi bit EEPE =0 sẽ bắt đầu hoạt động đọc EEPROM . Nội dung của ô nhớ EEPROM sẽ được đưa vào thanh ghi EEDR. Khi EEPROM được đọc thì CPU bị treo 4 chu kỳ xung nhịp trước khi thực hiện lệnh tiếp theo.

Để đọc EEPROM, cần theo các bước sau:

1. Chờ đến khi bit EEPM =0.
2. Ghi địa chỉ EEPROM vào thanh ghi EEAR.
3. Đặt bit EERE=1.
4. Đọc dữ liệu EEPROM trong thanh ghi EEDR.

### 2.5.5 Các thanh ghi I/O đa dụng (General Purpose I/O Registers)

Các thanh ghi này được dùng để chứa bất kỳ thông tin nào, đặc biệt hữu ích để chứa các biến toàn cục và các cờ trạng thái. Có 3 thanh ghi I/O đa dụng:

- **Thanh ghi I/O đa dụng 2 – GPIO2 (General Purpose I/O Register 2)**
- **Thanh ghi I/O đa dụng 1 – GPIO1 (General Purpose I/O Register 1)**
- **Thanh ghi I/O đa dụng 0 – GPIO0 (General Purpose I/O Register 0)**

Bit	7	6	5	4	3	2	1	0	
0x2B (0x4B)	MSB								LSB
Read/Write	R/W	GPIO2							
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x2A (0x4A)	MSB								LSB
Read/Write	R/W	GPIO1							
Initial Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
0x1E (0x3E)	MSB								LSB
Read/Write	R/W	GPIO0							
Initial Value	0	0	0	0	0	0	0	0	

Hình 2.20: Các thanh ghi đa dụng GPOIR2,GPIO1,GPIO0

### 2.5.6 Các thanh ghi bộ định thời (Timer/Counter Register)

ATmega324P có 3 bộ định thời (Timer/Counter) có nhiều chức năng mạnh. Bao gồm Timer/Counter0 8 bit, Timer/Counter1 16 bit và Timer/Counter2 8 bit. Hoạt động của các bộ định thời được điều khiển bởi các thanh ghi Timer. Các thanh ghi này được tìm hiểu kỹ trong chương 7.

**Bảng 2.8:** Tóm tắt các thanh ghi bộ định thời

Bộ Timer	Địa chỉ		Tên Thanh ghi	Chức năng
	Bộ nhớ	I/O		
Timer0	0x44	0x22	<b>TCCR0A</b>	Điều khiển Timer0 A
	0x45	0x25	<b>TCCR0B</b>	Điều khiển Timer0 B
	0x46	0x26	<b>TCNT0</b>	Bộ đếm Timer0
	0x47	0x27	<b>OCR0A</b>	So sánh ngõ ra Timer0 A
	0x48	0x28	<b>OCR0B</b>	So sánh ngõ ra Timer0 B
	0x35	0x15	<b>TIFR0</b>	Cờ báo ngắt Timer0
	0x6E		<b>TIMSK0</b>	Che ngắt Timer0
Timer1	0x80		<b>TCCR1A</b>	Điều khiển Timer1 A
	0x81		<b>TCCR1B</b>	Điều khiển Timer1 B
	0x82		<b>TCCR1C</b>	Điều khiển Timer1 C
	0x84		<b>TCNT1L</b>	Bộ đếm Timer1 byte thấp
	0x85		<b>TCNT1H</b>	Bộ đếm Timer1 byte cao
	0x86		<b>ICR1L</b>	Bắt ngõ vào Timer1 byte thấp
	0x87		<b>ICR1H</b>	Bắt ngõ vào Timer1 byte cao
	0x88		<b>OCR1AL</b>	So sánh ngõ ra Timer1 A byte thấp
	0x89		<b>OCR1AH</b>	So sánh ngõ ra Timer1 A byte cao
	0x8A		<b>OCR1BL</b>	So sánh ngõ ra Timer1 B byte thấp
	0x8B		<b>OCR1BH</b>	So sánh ngõ ra Timer1 B byte cao
	0x36	0x16	<b>TIFR1</b>	Cờ báo ngắt Timer1
Timer2	0xB0		<b>TCCR2A</b>	Điều khiển Timer2 A
	0xB1		<b>TCCR2B</b>	Điều khiển Timer2 B
	0xB2		<b>TCNT2</b>	Bộ đếm Timer2
	0xB3		<b>OCR2A</b>	So sánh ngõ ra Timer2 A
	0xB4		<b>OCR2B</b>	So sánh ngõ ra Timer2 B
	0xB6		<b>ASSR</b>	Trạng thái bắt đồng bộ
	0x37	0x17	<b>TIFR2</b>	Cờ báo ngắt Timer2
	0x70		<b>TIMSK2</b>	Che ngắt Timer2
	0x43	0x23	<b>GTCCR</b>	Điều khiển Timer chung

### 2.5.7 Các thanh ghi cổng nối tiếp (Serial Port Registers)

ATmega324P có thể truyền nối tiếp với nhiều phương thức truyền SPI, USART, TWI giúp làm việc linh hoạt với các ngoại vi. Các thanh ghi điều khiển cổng nối tiếp cũng sẽ được trình bày trong chương 8.

**Bảng 2.9:** Tóm tắt các thanh ghi cổng nối tiếp

Cổng nối tiếp	Địa chỉ		Tên Thanh ghi	Chức năng
	Bộ nhớ	I/O		
SPI	0x4C	0x2C	<b>SPCR0</b>	Điều khiển SPI
	0x4D	0x2D	<b>SPSR0</b>	Trạng thái SPI
	0x4E	0x2E	<b>SPDR0</b>	Dữ liệu SPI

USART0	0xC0	<b>UCSR0A</b>	Trạng thái và điều khiển USART0 A
	0xC1	<b>UCSR0B</b>	Trạng thái và điều khiển USART0 B
	0xC2	<b>UCSR0C</b>	Trạng thái và điều khiển USART0 C
	0xC4	<b>UBRR0L</b>	Tốc độ baud USART0 byte thấp
	0xC5	<b>UBRR0H</b>	Tốc độ baud USART0 byte cao
	0xC6	<b>UDR0</b>	Dữ liệu xuất/nhập USART0
USART1	0xC8	<b>UCSR1A</b>	Trạng thái và điều khiển USART1 A
	0xC9	<b>UCSR1B</b>	Trạng thái và điều khiển USART1 B
	0xCA	<b>UCSR1C</b>	Trạng thái và điều khiển USART1 C
	0xCC	<b>UBRR1L</b>	Tốc độ baud USART1 byte thấp
	0xCD	<b>UBRR1H</b>	Tốc độ baud USART1 byte cao
	0xCE	<b>UDR1</b>	Dữ liệu xuất/nhập USART1
TWI	0xB8	<b>TWBR</b>	Tốc độ bit TWI
	0xB9	<b>TWSR</b>	Trạng thái TWI
	0xBA	<b>TWAR</b>	Địa chỉ (slave) TWI
	0xBB	<b>TWDR</b>	Dữ liệu TWI
	0xBC	<b>TWCR</b>	Điều khiển TWI
	0xBD	<b>TWAMR</b>	Che địa chỉ (slave) TWI

### 2.5.8 Các thanh ghi bộ ADC và bộ so sánh tương tự

ATmega324P có bộ chuyển đổi ADC 10 bit và bộ so sánh tương tự. Việc điều khiển hoạt động của ADC và bộ so sánh cũng dùng các thanh ghi sẽ được xem xét trong chương 9.

Bảng 2.10: Tóm tắt các thanh ghi ADC và bộ so sánh

	<b>Địa chỉ</b>		<b>Tên Thanh ghi</b>	<b>Chức năng</b>
	<b>Bộ nhớ</b>	<b>I/O</b>		
Bộ ADC	0x78		<b>ADCL</b>	Dữ liệu ADC byte thấp
	0x79		<b>ADCH</b>	Dữ liệu ADC byte cao
	0x7A		<b>ADCSRA</b>	Trạng thái và điều khiển ADC A
	0x7B		<b>ADCSR</b>	Trạng thái và điều khiển ADC B
	0x7C		<b>ADMUX</b>	Chọn kênh ADC
	0x7E		<b>DIDR0</b>	Cảm ngõ vào số 0
Bộ so sánh tương tự AC	0x50	0x30	<b>ACSR</b>	Trạng thái và điều khiển bộ so sánh
	0x7F		<b>DIDR1</b>	Cảm ngõ vào số 1

### 2.5.9 Các thanh ghi ngắt (Interrupt Registers)

Ngắt là yêu cầu từ ngoại vi, vì vậy ngắt của ATmega324P có ngắt từ các ngoại vi bên trong như bộ định thời, cổng nối tiếp, bộ biến đổi ADC, .. và các ngắt từ các thiết bị ngoài qua các chân I/O. Các thanh ghi điều khiển ngắt cũng sẽ được trình bày trong chương 10.

Bảng 2.11: Tóm tắt các thanh ghi ngắt

<b>Địa chỉ</b>		<b>Tên Thanh ghi</b>	<b>Chức năng</b>
<b>Bộ nhớ</b>	<b>I/O</b>		
0x55	0x35	<b>MCUCR</b>	Điều khiển MCU
0x69		<b>EICRA</b>	Điều khiển ngắt ngoài
0x3C	0x1C	<b>EIFR</b>	Cờ báo ngắt ngoài

0x3D	0x1D	<b>EIMSK</b>	Che ngắt ngoài
0x68		<b>PCICR</b>	Điều khiển ngắt Pin Change
0x3B	0x1B	<b>PCIFR</b>	Cờ báo ngắt Pin Change
0x6B		<b>PCMSK0</b>	Che ngắt Pin Change 0
0x6C		<b>PCMSK1</b>	Che ngắt Pin Change 1
0x6D		<b>PCMSK2</b>	Che ngắt Pin Change 2
0x73		<b>PCMSK3</b>	Che ngắt Pin Change 3

### 2.5.10 Các thanh ghi hệ thống

Các thanh ghi này có liên quan các phần cứng hệ thống như bộ dao động, mạch reset và các chế độ giám sát nguồn cùng nhiều chế độ nghỉ. Tất cả sẽ được trình bày riêng trong các phần tiếp theo trong chương này.

#### Câu hỏi ôn tập

1. Thanh ghi SREG có bao nhiêu bit?
2. Thực thi lệnh ADD R16, R17. Hãy cho biết giá trị của các cờ trong thanh ghi trạng thái SREG thay đổi ra sao khi thực hiện xong lệnh, với nội dung thanh ghi của R16 và R17 lần lượt trong các trường hợp sau:
  - a. 0x9F và 0x61
  - b. 0x75 và 0x6A
  - c. 0x16 và 0xEA
3. Cho biết giá trị đầu của thanh ghi SP là bao nhiêu? Tại sao?
4. Thanh ghi SP có bao nhiêu bit? Tại sao?
5. Cấu trúc LIFO (Last In First Out) của ngăn xếp là như thế nào?
6. Với SP = \$0700, khi thực hiện xong lệnh PUSH R20, thì SP thay đổi ra sao?
7. Cho biết chức năng của các thanh ghi DDRx, PORTx, PINx (với x là A, B, C, D)
8. Cho biết ý nghĩa của đoạn chương trình sau:

```
LDI R16, 0x0F
OUT DDRB, R16
```

9. Viết đoạn chương trình xuất giá trị \$0F ra Port A.
10. Viết đoạn chương trình nhập dữ liệu từ Port D và cất vào thanh ghi R5

## 2.6 XUNG NHỊP HỆ THỐNG (SYSTEM CLOCK)

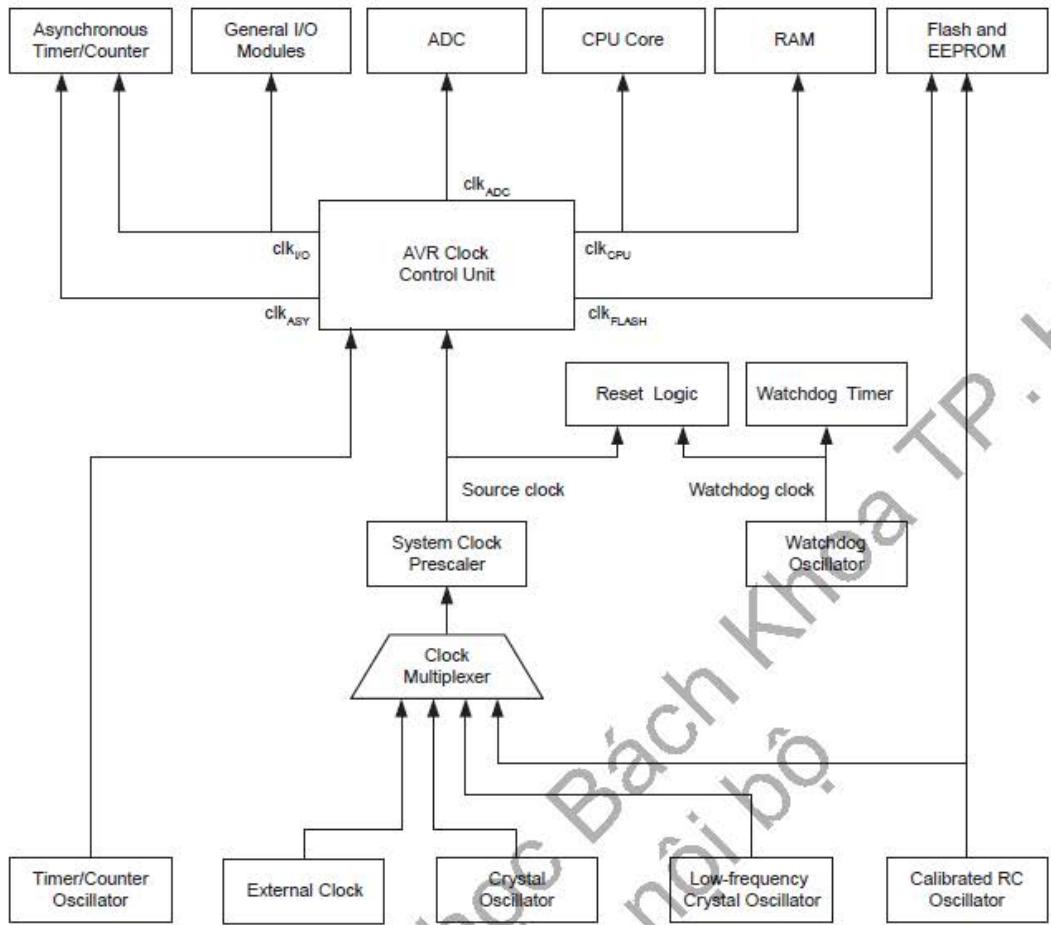
### 2.6.1 Đặc điểm hệ thống xung nhịp

#### 2.6.1.1 Hệ thống xung nhịp

Xung nhịp (Clock) cần thiết cho các hoạt động cơ bản của 1 hệ thống vi điều khiển: chu kỳ thực hiện lệnh, định thời truy xuất bộ nhớ, giao tiếp ngoại vi ... Tần số xung nhịp quyết định tốc độ thực thi các lệnh trong chương trình.

AVR có nhiều loại xung nhịp cho CPU và các thành phần ngoại vi bên trong. Các xung nhịp này không cần thiết hoạt động cùng lúc. Để tiết kiệm điện năng, xung nhịp thành phần nào không dùng thì có thể treo hay chuyển sang chế độ ngủ.

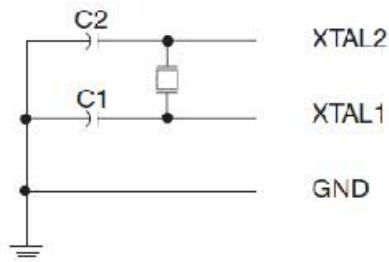
- **clkCPU** xung nhịp này cung cấp cho hoạt động của lõi AVR.
- **clkIO** xung nhịp này cung cấp cho ngoại vi như bộ định thời và giao tiếp nối tiếp.
- **clkFLASH** xung nhịp này cung cấp cho giao tiếp bộ nhớ Flash.
- **clkASY** xung nhịp này cung cấp cho bộ định thời bắt đồng bộ.
- **clkADC** xung nhịp này cung cấp cho bộ biến đổi ADC.



Hình 2.21: Phân bổ các xung nhịp.

### 2.6.1.2 Chân nối bộ dao động thạch anh

Có 2 chân kết nối XTAL1 và XTAL2 với bộ dao động thạch anh. Ngõ ra XTAL2 có thể dùng làm nguồn xung nhịp cho các thiết bị bên ngoài. Có thể thay dao động thạch anh bằng bộ cộng hưởng gồm:



Hình 2.22: Kết nối dao động thạch anh

Tụ điện C1 và C2 có giá trị bằng nhau.

### 2.6.1.3 Ngõ ra xung nhịp (Clock Output)

ATmega324P có thể xuất xung nhịp hệ thống qua chân CLKO (chân PB1). Để kích hoạt ngõ ra xung nhịp, cầu chì CKOUT phải được lập trình (xem mục 2.9.1). Chế độ này phù hợp khi xung nhịp được sử dụng để điều khiển các mạch khác trong hệ thống. Bất kỳ nguồn xung nhịp nào, bao gồm cả bộ dao động RC bên trong, đều có thể được chọn xuất ra chân

CLKO. Nếu sử dụng bộ chia tỷ lệ xung nhịp, thì nó sẽ là đầu ra của xung nhịp hệ thống đã chia.

#### 2.6.1.4 Trình tự khởi động xung nhịp (Clock Start-up Sequence)

Bất kỳ nguồn xung nhịp nào cũng cần có Vcc vừa đủ để bắt đầu dao động và một số chu kỳ dao động tối thiểu trước khi nó có thể được coi là ổn định.

Để đảm bảo Vcc vừa đủ, có thêm khởi động (reset) nội với độ trễ thời gian chờ ( $t_{TOUT}$  - time-out delay) sau tất cả các nguồn khởi động khác kết thúc. Độ trễ ( $t_{TOUT}$ ) được tính từ bộ tạo dao động giám sát Watchdog và một số chu kỳ trễ được đặt bởi các bit cầu chì SUT1..0 và CKSEL0. Để tìm hiểu chi tiết hơn về vấn đề này, cần đọc thêm data sheet của ATmega324P.

#### 2.6.2 Các nguồn xung nhịp

ATmega324P có nhiều nguồn xung nhịp khác nhau. Xung nhịp từ nguồn dao động thạch anh, dao động RC bên trong hay nguồn dao động ngoài. Nguồn xung nhịp để đưa đến hệ thống được lựa chọn bằng các bit cầu chì CKSEL3..0 được trình bày ở mục 2.9.1.

**Bảng 2.12** Tùy chọn nguồn xung nhịp

Tùy chọn xung nhịp thiết bị (Device Clocking Option)	CKSEL3..0
Bộ dao động thạch anh công suất thấp (Low Power Crystal Oscillator)	1111 - 1000
Bộ dao động thạch anh toàn tầm (Full Swing Crystal Oscillator)	0111 - 0110
Bộ dao động thạch anh tần số thấp (Low Frequency Crystal Oscillator)	0101 - 0100
Bộ dao động RC nội 128kHz (Internal 128kHz RC Oscilltor)	0011
Bộ dao động RC nội hiệu chỉnh được (Calibrated Internal RC Oscillator)	0010
Xung nhịp ngoài (External Clock)	0000
Dự trữ (Reserved)	0001

##### 2.6.2.1 Bộ dao động thạch anh công suất thấp (Low Power Crystal Oscillator)

Dao động thạch anh này là bộ tạo dao động công suất thấp, với giảm thiểu điện áp ở ngõ ra XTAL2 làm cho mức tiêu thụ điện năng là thấp nhất. Nhưng không có khả năng tạo nguồn xung cho các I/O khác bên ngoài chip khi cần xung nhịp.

Bộ tạo dao động công suất thấp có thể hoạt động ở ba chế độ khác nhau, mỗi chế độ được tối ưu hóa cho một dải tần số cụ thể. Các chế độ này được chọn bởi các cầu chì CKSEL3..1.

**Bảng 2.13:** Các chế độ dao động thạch anh công suất thấp

Dải tần số (MHz)	CKSEL3..1	Giá trị tụ điện để xuất C1 và C2 (pF)
0.4 – 0.9	100	-
0.9 – 3.0	101	12 – 22
3.0 – 8.0	110	12 – 22
8.0 – 16.0	111	12 – 22

Chú ý trường hợp dải tần số 0.4MHz – 0.9Mhz (CKSEL3..1 = 100) không dùng thạch anh mà chỉ dành cho bộ cộng hưởng gồm.

##### 2.6.2.2 Bộ dao động thạch anh toàn tầm (Full Swing Crystal Oscillator)

Bộ dao động này có mức điện áp tối đa ở ngõ ra XTAL2. Bộ dao động này có ích trong việc tạo nguồn xung nhịp cho các I/O bên ngoài và khi làm việc trong môi trường nhiễu.

Chú ý là bộ dao động Full Swing hoạt động với điện áp Vcc từ 2.7V đến 5.5V.

**Bảng 2.14** Dao động thạch anh toàn tần

Dải tần số (MHz)	CKSEL3..1	Giá trị tụ điện đề xuất C1 và C2 (pF)
0.4 – 20.0	011	12 – 22

#### 2.6.2.3 Bộ dao động thạch anh tần số thấp (Low Frequency Crystal Oscillator)

Để thực hiện các mạch cần tiêu thụ công suất thấp thì xung nhịp cần ở tần số thấp. Bộ dao động thạch anh tần số thấp được tối ưu hóa để sử dụng với thạch anh đồng hồ 32.768 KHz. Khi chọn thạch anh, cần chú ý đến điện dung tải và trở kháng nối tiếp tương đương của thạch anh (ESR - Equivalent Series Resistance). Cả hai giá trị đều được chỉ định bởi nhà cung cấp thạch anh.

#### 2.6.2.4 Bộ dao động RC nội 128 KHz (128KHz Internal Oscillator)

Bộ dao động 128 KHz có công suất thấp và dành cho hệ thống không cần độ chính xác cao. Để chọn bộ dao động này thì cầu chì CKSEL3..0 = 0011.

#### 2.6.2.5 Bộ dao động RC nội hiệu chỉnh được (Calibrated Internal RC Oscillator)

Bộ dao động RC nội được dùng với tần số là 8 MHz khi thiết lập cầu chì CKSEL3..0 = 0010.Thêm với cầu chì CKDIV8 được lập trình (xem mục 2.9.1), bộ chia 8 tần số được thực hiện, xung nhịp hệ thống sẽ là 1 MHz. Đây là nguồn xung nhịp danh định của ATmeg324P khi xuất xưởng. Lúc này hệ thống xung nhịp hoạt động mà không cần bất kỳ thành phần nào bên ngoài.

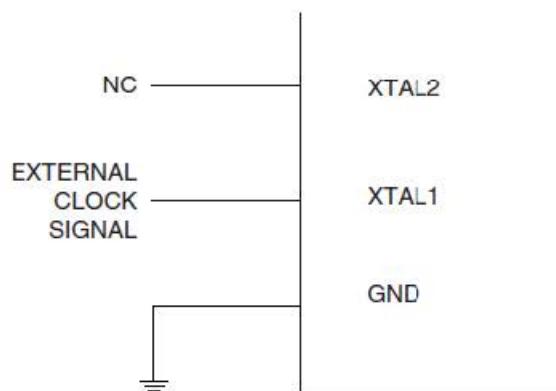
Do bộ dao động RC phụ thuộc vào nguồn điện áp và nhiệt độ, tần số có thể thay đổi từ 7.3 MHz đến 8.1 MHz. Vì vậy cần hiệu chỉnh để có tần số chính xác 8 MHz, dùng thanh ghi OSCCAL (Oscillator Calibration Register).

Bit	7	6	5	4	3	2	1	0	OSCCAL
(0x66)	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	
Read/Write	R/W								
Initial Value									Device Specific Calibration Value

**Hình 2.23:** Thanh ghi OSCCAL hiệu chỉnh tần số dao động nội

#### 2.6.2.6 Xung nhịp ngoài (External Clock)

Khi CKSEL3..0 = 0000, một nguồn xung ngoài có tần số tối đa 20 MHz có thể dùng trong hệ thống.



**Hình 2.24:** Xung nhịp ngoài đưa vào MCU

### 2.6.3 Bộ tỉ lệ xung nhịp hệ thống (System Clock Prescaler)

Như hình 2.21, ta thấy xung nhịp hệ thống sau khi lựa chọn được đưa qua 1 bộ chia tần số. Tính năng này có thể được sử dụng để giảm tần số xung nhịp hệ thống và mức tiêu thụ điện năng khi hệ thống có yêu cầu về công suất xử lý thấp.

Bộ chia tần số được chọn lựa theo những tỉ lệ khác nhau, được thiết lập qua thanh ghi CLKPR(Clock Prescaler Register)

Bit (0x61)	7	6	5	4	3	2	1	0	CLKPR
Read/Write	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0	
Initial Value	R/W	R	R	R	R/W	R/W	R/W	R/W	
	0	0	0	0	See Bit Description				

Hình 2.25: Thanh ghi CLKPR thiết lập tỉ lệ chia tần số xung nhịp

- Bit 7 – CLKPCE: Cho phép thay đổi bộ chia xung nhịp (Clock Prescaler Change Enable)  
Bit CLKPCE phải được ghi giá trị 1 để cho phép. Bit CLKPCE chỉ được cập nhật khi các bit còn lại là 0. Nghĩa là ghi vào thanh ghi CLKPR giá trị 0x80. Bit CLKPCE sẽ được xóa bởi phần cứng sau bốn chu kỳ sau khi ghi.
- Bit 3:0 – CLKPSn: Chọn tỉ lệ chia xung nhịp (Clock Prescaler Select)  
Các bit này xác định hệ số chia giữa nguồn xung nhịp và xung nhịp hệ thống bên trong.  
Các hệ số chia được chọn theo như bảng dưới đây.

Bảng 2.15: Chọn lựa hệ số chia

CLKPS3	CLKPS2	CLKPS1	CLKPS0	Hệ số chia clock (Clock Division Factor)
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Dự trữ
1	0	1	0	Dự trữ
1	0	1	1	Dự trữ
1	1	0	0	Dự trữ
1	1	0	1	Dự trữ
1	1	1	0	Dự trữ
1	1	1	1	Dự trữ

Chú ý là khi ghi hệ số cần chọn vào thanh ghi CLKPR, thì bit CLKPCE phải ghi giá trị 0 (ví dụ chọn bộ chia 8 thì giá trị ghi vào thanh ghi CLKPR là 0x03). Và việc ghi giá trị này phải được thực hiện sau khi cho bit CLKPCE giá trị 1 trước đó trong vòng 4 chu kỳ. Nếu không việc chia xung nhịp đã lựa chọn sẽ không được thực hiện.

## Câu hỏi ôn tập

1. Bộ dao động danh định nào được chọn khi sản xuất? Tại sao?
2. Khi đó chu kỳ lệnh của CPU là bao nhiêu? Tại sao?
3. Nếu sử dụng thạch anh bên ngoài là 16MHZ và sử dụng bộ tỉ lệ là bao nhiêu để xung nhịp hệ thống có tần số là 1MHz.
4. Để có tỉ lệ chia tần số là 64, giá trị ghi vào thanh ghi CLKPR là bao nhiêu?
5. Muốn chân CLKO (chân PB1) làm nguồn xung nhịp thì phải làm gì?

## 2.7 KIỂM SOÁT NGUỒN VÀ CÁC CHẾ ĐỘ NGỦ

Đối với 1 số hệ thống vi điều khiển, việc kiểm soát nguồn giúp cho các yêu cầu thiết kế cần tiết kiệm năng lượng. AVR có nhiều cách để giảm tiêu hao năng lượng như là cho sử dụng các chế độ ngủ (Sleep Modes), tắt bộ phát hiện sụt áp BOD, ...

### 2.7.1 Các chế độ ngủ (Sleep Modes)

Chế độ ngủ cho phép ứng dụng tắt các khối không sử dụng trong MCU, do đó tiết kiệm năng lượng. AVR cung cấp các chế độ ngủ khác nhau cho phép người dùng điều chỉnh mức tiêu thụ năng lượng theo yêu cầu của ứng dụng.

ATmega324P có 6 chế độ ngủ: nghỉ (Idle), giảm nhiễu ADC (ADC Noise Reduction), tắt nguồn (Power-down), tiết kiệm nguồn (Power-save), chờ (Standby), và chờ mở rộng (Extended Standby). Chế độ ngủ sẽ được thực thi bằng lệnh SLEEP.

Khi MCU vào chế độ ngủ, một ngắt được kích hoạt xảy ra làm MCU bị đánh thức (wake-up). MCU thực hiện quá trình ngắt và tiếp tục thực hiện lệnh sau SLEEP. Nội dung của tập thanh ghi và SRAM không thay đổi khi thiết bị thức dậy từ chế độ ngủ. Nếu có reset xảy ra trong chế độ ngủ, MCU cũng sẽ đánh thức và thực thi từ địa chỉ reset.

#### 2.7.1.1 Chế độ nghỉ (Idle Mode)

Ở chế độ này, CPU ngưng hoạt động nhưng các ngoại vi (SPI, USART, bộ so sánh tương tự, ADC, giao tiếp 2 dây TWI, bộ định thời/bộ đếm) và hệ thống ngắt tiếp tục hoạt động. Chế độ ngủ này về cơ bản tạm dừng  $\text{clk}_{\text{CPU}}$  và  $\text{clk}_{\text{FLASH}}$ , trong khi cho phép các xung nhịp khác vẫn chạy (xem các kiểu xung nhịp hình 2.21).

Chế độ nghỉ cho phép MCU bị đánh thức từ các ngắt ngoài cũng như các ngắt nội.

#### 2.7.1.2 Chế độ giảm nhiễu ADC (ADC Noise Reduction Mode)

Ở chế độ này, CPU ngưng hoạt động nhưng các ngoại vi (SPI, bộ so sánh tương tự, ADC, giao tiếp 2 dây TWI, bộ định thời/bộ đếm2, ngắt ngoài) và hệ thống giám sát watchdog tiếp tục hoạt động. Chế độ ngủ này về cơ bản tạm dừng  $\text{clk}_{\text{CPU}}$ ,  $\text{clk}_{\text{FLASH}}$  và  $\text{clk}_{\text{I/O}}$ , trong khi cho phép các xung nhịp khác vẫn chạy (xem các kiểu xung nhịp hình 2.21). Điều này cải thiện môi trường nhiễu cho ADC, cho phép các phép đo có độ phân giải cao hơn. Nếu ADC được cho phép, quá trình chuyên đổi sẽ tự động bắt đầu khi vào chế độ này.

Một trong các điều kiện để hệ thống thoát khỏi chế độ ngủ này:

- Các ngắt: ADC, ngắt ngoài, PCINT7:4 (chỉ ngắt mức), Pin change, Timer2, SPM/EEPROM, watchdog.
- Giao tiếp 2 dây TWI.
- Các reset: ngoài, watchdog và Brown-out.

#### 2.7.1.3 Chế độ tắt nguồn (Power-down Mode)

Ở chế độ này, bộ dao động bên ngoài bị dừng, do đó các xung clk đều dừng. Chỉ có các ngắt ngoài, giao tiếp 2 dây TWI và hệ thống giám sát watchdog tiếp tục hoạt động. Chế độ này về cơ bản sẽ tạm dừng tất cả các thành phần xung nhịp, chỉ cho phép hoạt động không đồng bộ.

Một trong các điều kiện để hệ thống thoát khỏi chế độ ngủ này:

- Các ngắt: ngắt ngoài, PCINT7:4 (chỉ ngắt mức), Pin change, watchdog.
- Giao tiếp 2 dây TWI.
- Các reset: ngoài, watchdog và Brown-out.

#### 2.7.1.4 Chế độ tiết kiệm nguồn (Power-save Mode)

Chế độ ngủ giống với chế độ tắt nguồn, nhưng khác là khi cho phép Timer2 hoạt động thì nó vẫn chạy trong chế độ ngủ. Khi đó hệ thống sẽ được đánh thức từ ngắt Timer2.

Nếu không sử dụng Timer2, thì nên sử dụng chế độ tắt nguồn.

#### 2.7.1.5 Chế độ chờ (Standby Mode)

Chế độ ngủ giống với chế độ tắt nguồn, nhưng khác là có sử dụng nguồn xung nhịp từ thạch anh hay bộ cộng hưởng gồm bên ngoài.

#### 2.7.1.6 Chế độ chờ mở rộng (Extended Standby Mode)

Chế độ ngủ giống với chế độ tiết kiệm nguồn, nhưng khác là có sử dụng các bộ dao động.

**Bảng 2.16** Tóm tắt hoạt động của các chế độ ngủ

Chế độ ngủ (Sleep Mode)	Vùng clock hoạt động					Bộ dao động	Nguồn đánh thức (wake-up)						Cảm BDO mềm		
	clk <sub>CPU</sub>	clk <sub>FLASH</sub>	clk <sub>I/O</sub>	clk <sub>ADC</sub>	clk <sub>ASY</sub>		Cho phép nguồn xung nhịp chính	Cho phép dao động Timer	INT2..1 và Pin Change	Khớp địa chỉ TWI	Timer2	SPM/EEPROM	ADC	Ngắt WDT	I/O khác
Nghỉ (Idle)			X	X	X	X	X <sup>(2)</sup>	X	X	X	X	X	X	X	X
Giảm nhiễu ADC (ADCNRM)			X	X	X	X	X <sup>(2)</sup>	X <sup>(3)</sup>	X	X <sup>(2)</sup>	X	X	X	X	
Tắt nguồn (Power-down)								X <sup>(3)</sup>	X					X	X
Tiết kiệm nguồn (Power-safe)					X		X <sup>(2)</sup>	X <sup>(3)</sup>	X	X				X	X
Chờ (Standby) <sup>(1)</sup>							X		X <sup>(3)</sup>	X				X	X
Chờ mở rộng (Extended Standby)					X <sup>(2)</sup>	X	X <sup>(2)</sup>	X <sup>(3)</sup>	X	X				X	X

Chú thích: (1) nguồn xung nhịp từ thạch anh hay bộ cộng hưởng gồm bên ngoài.

(2) Timer2 hoạt động chế độ bất đồng bộ.

(3) Ngắt ngoài INT0 là ngắt mức.

### Thanh ghi điều khiển chế độ ngủ SMCR (Sleep Mode Control Register)

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	-	-	-	-	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Hình 2.26: Thanh ghi SMCR điều khiển chế độ ngủ

- Bit 3:1 – SMn: chọn chế độ ngủ (Sleep Mode n)  
Các bit để chọn 1 trong 6 chế độ ngủ theo bảng 2.10.

Bảng 2.17: Chọn chế độ ngủ

SM2	SM1	SM0	Chế độ ngủ (Sleep Mode)
0	0	0	Nghỉ (Idle)
0	0	1	Giảm nhiễu ADC (ADC Noise Reduction)
0	1	0	Tắt nguồn (Power-down)
0	1	1	Tiết kiệm nguồn (Power-safe)
1	0	0	Dự trữ
1	0	1	Dự trữ
1	1	0	Chờ (Standby)
1	1	1	Chờ mở rộng (Extended Standby)

- Bit 0 – SE: cho phép chế độ ngủ (Sleep Enable)

Bit SE phải được ghi giá trị 1 để làm cho MCU chuyển sang chế độ nghỉ khi lệnh SLEEP được thực thi. Việc ghi này nên thực hiện ngay trước khi thực hiện lệnh SLEEP và xóa nó ngay sau khi MCU được đánh thức.

#### 2.7.2 Bộ phát hiện BOD (Brown-Out Detector)

AVR có bộ phát hiện sụt áp nguồn Brown-out (Brown Out Detector - BOD) sẽ chủ động giám sát điện áp nguồn trong thời gian ngủ. Bộ BOD được cho phép bằng các cầu chì BODLEVEL (xem mục 2.9.3). Khi điện áp Vcc thấp, dữ liệu EEPROM có thể bị hỏng do điện áp cung cấp quá thấp để CPU và EEPROM hoạt động bình thường. Bộ BOD có nhiệm vụ phát hiện sụt áp dưới ngưỡng cho phép, nếu có thì sẽ reset CPU. Khi Vcc tăng trở lại trên mức kích hoạt, bộ BOD thoát khỏi trạng thái reset. CPU bắt đầu làm việc sau khi chu kỳ time-out kết thúc.

Để tiết kiệm năng lượng, có thể tắt BOD bằng phần mềm đối với một số chế độ ngủ. Nếu BOD bị cấm bằng phần mềm, chức năng BOD sẽ bị cấm ngay sau khi vào chế độ ngủ. Khi bị đánh thức, BOD sẽ tự động được cho phép trở lại.

Trong thanh ghi MCUCR có các bit điều khiển cấm/cho phép bộ BOD.

### Thanh ghi điều khiển MCU - MCUCR (MCU Control Register)

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	JTD	BODS	BODSE	PUD	-	-	IVSEL	IVCE	MCUCR
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

## Hình 2.27: Thanh ghi MCUCR cho phép/cấm BOD

- Bit 6 – **BODS**: ngủ BOD (BOD Sleep)

Bit BODS phải được ghi giá trị 1 để tắt BOD trong chế độ ngủ. Việc ghi vào bit BODS được điều khiển bởi một trình tự thời gian và bit cho phép BODSE. Để tắt BOD trong các chế độ ngủ liên quan, cả hai bit BODS và BODSE trước tiên phải được ghi giá trị 1. Sau đó để thiết lập bit BODS, BODS phải được ghi giá trị 1 và BODSE phải được ghi giá trị 0 trong vòng bốn chu kỳ xung nhịp.

Bit BODS tích cực ba chu kỳ xung nhịp sau khi nó được thiết lập. Lệnh SLEEP phải được thực hiện trong khi BODS đang tích cực để tắt BOD cho chế độ ngủ thực tế. Bit BODS sẽ tự động bị xóa sau ba chu kỳ xung nhịp.

- Bit 5 – **BODSE**: cho phép ngủ BOD (BOD Sleep Enable)

Bit BODSE cho phép thiết lập bit điều khiển BODS, như được giải thích trong phần mô tả bit BODS. Việc vô hiệu hóa BOD được kiểm soát bởi một trình tự thời gian.

### 2.7.3 Thanh ghi giảm năng lượng PRR (Power Reduction Register)

Thanh ghi giảm năng lượng (PRR) cung cấp một phương pháp dừng xung nhịp cho các ngoại vi riêng lẻ để giảm tiêu thụ điện năng. Trạng thái hiện tại của ngoại vi bị đóng băng và không thể đọc hoặc ghi các thanh ghi I/O. Các tài nguyên được sử dụng bởi ngoại vi khi dừng xung nhịp sẽ vẫn bị chiếm dụng. Do đó, trong hầu hết các trường hợp, ngoại vi phải bị cấm trước khi dừng xung nhịp. Việc đánh thức ngoại vi, được thực hiện bằng cách xóa bit tương ứng trong PRR, làm cho nó ở trạng thái giống như trước khi tắt.

Chế độ tắt ngoại vi có thể được sử dụng ở chế độ nghỉ và chế độ hoạt động để giảm đáng kể mức tiêu thụ năng lượng tổng thể. Trong tất cả các chế độ ngủ khác, xung nhịp đã bị dừng.

#### Thanh ghi giảm năng lượng 0 – PRR0 (Power Reduction Register)

Bit	7	6	5	4	3	2	1	0	
(0x64)	PRTWI	PRTIM2	PRTIMO	PRUSART1	PRTIM1	PRSPI0	PRUSART0	PRADC	PRR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Hình 2.28: Thanh ghi PRR0 điều khiển giảm năng lượng

- Bit 7 – **PRTWI**: giảm năng lượng TWI (Power Reduction TWI)

Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động giao tiếp 2 dây TWI bằng cách dừng xung nhịp đến nó. Khi đánh thức TWI một lần nữa, TWI nên được khởi tạo lại để đảm bảo hoạt động bình thường.

- Bit 6 – **PRTIM2**: giảm năng lượng Timer2 (Power Reduction Timer /Counter2)

Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động bộ Timer2 ở chế độ đồng bộ. Khi bộ Timer2 được cho phép, hoạt động sẽ tiếp tục như trước khi ngưng.

- Bit 5 – **PRTIM0**: giảm năng lượng Bộ Timer0 (Power Reduction Timer /Counter0)

Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động bộ Timer0. Khi bộ Timer0 được cho phép, hoạt động sẽ tiếp tục như trước khi ngưng.

- Bit 4 – **PRUSART1**: giảm năng lượng USART1 (Power Reduction USART1)  
Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động USART1 bằng cách dừng xung nhịp đến nó. Khi đánh thức lại USART1, USART1 nên được khởi tạo lại để đảm bảo hoạt động bình thường.
- Bit 3 – **PRTIM1**: giảm năng lượng Bộ Timer1 (Power Reduction Timer /Counter1)  
Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động bộ Timer1. Khi bộ Timer1 được cho phép, hoạt động sẽ tiếp tục như trước khi ngưng.
- Bit 2 – **PRSPI0**: giảm năng lượng SPI (Power Reduction SPI)  
Việc ghi giá trị vào bit này sẽ ngưng hoạt động bộ giao tiếp ngoại vi nối tiếp SPI bằng cách dừng xung nhịp đến nó. Khi đánh thức SPI một lần nữa, SPI nên được khởi tạo lại để đảm bảo hoạt động bình thường.
- Bit 1 – **PRUSART0**: giảm năng lượng USART0 (Power Reduction USART0)  
Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động USART0 bằng cách dừng xung nhịp đến nó. Khi đánh thức lại USART0, USART0 nên được khởi tạo lại để đảm bảo hoạt động bình thường.
- Bit 0 – **PRADC**: giảm năng lượng ADC (Power Reduction ADC)  
Việc ghi giá trị 1 vào bit này sẽ ngưng hoạt động bộ biến đổi ADC. Bộ ADC phải bị cấm trước khi tắt. Bộ so sánh tương tự không thể sử dụng bộ MUX đầu vào ADC khi bộ ADC tắt.

Để giảm thiểu mức tiêu thụ năng lượng, chế độ ngủ nên được sử dụng càng nhiều càng tốt. Tất cả các chức năng không cần thiết nên được tắt. Ngoài ra, còn các khả năng khác có thể cũng giảm mức tiêu thụ, tham khảo thêm data sheet của ATmega324P.

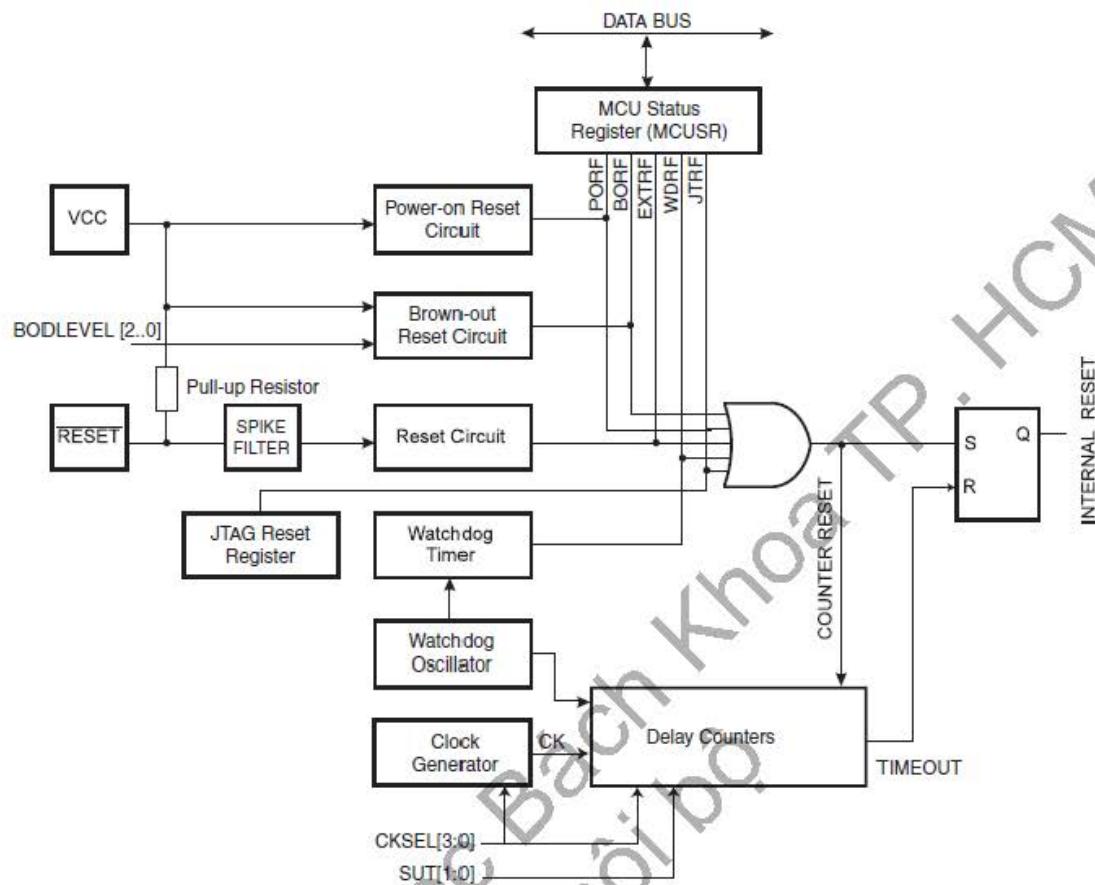
## 2.8 ĐIỀU KHIỂN HỆ THỐNG VÀ RESET

Trong khi reset, tất cả các thanh ghi I/O được đặt lại giá trị ban đầu của chúng và chương trình bắt đầu thực hiện từ địa chỉ reset (reset vector). Lệnh đặt tại địa chỉ này phải là lệnh nhảy tuyệt đối (JMP) đến reset chương trình xử lý. Nếu chương trình không sử dụng ngắn, các vector ngắn sẽ không được sử dụng và chương trình có thể được đặt tại các địa chỉ này. Trường hợp này cũng xảy ra nếu địa chỉ reset nằm trong phần ứng dụng (Application section) trong khi các vector ngắn nằm trong phần khởi động (Boot section) hoặc ngược lại.

Các cổng I/O của AVR ngay lập tức về trạng thái ban đầu khi nguồn reset hoạt động. Điều này không yêu cầu bất kỳ nguồn xung nhịp nào đang chạy.

Sau khi tắt cả các nguồn reset không hoạt động, bộ đếm tạo trễ sẽ được kích khởi, kéo dài quá trình reset bên trong. Điều này cho phép nguồn điện đạt mức ổn định trước khi bắt đầu hoạt động bình thường. Khoảng thời gian chờ  $t_{TOUT}$  (time-out) của bộ đếm tạo trễ được xác định bởi việc thiết lập các bit cầu chì SUT và CKSEL.

Sau khi reset, nội dung của đa số các thanh ghi thường được xóa. Ví dụ như tập thanh ghi hay thanh ghi PC = 0x0000. Các Port đều là ngõ vào do các thanh ghi DDRx = 0x00.



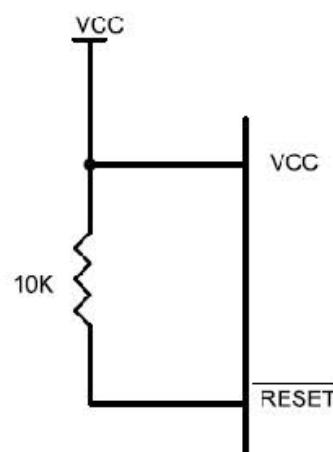
Hình 2.29: Logic khởi reset

### 2.8.1 Các nguồn Reset

ATmega324P có 5 nguồn reset:

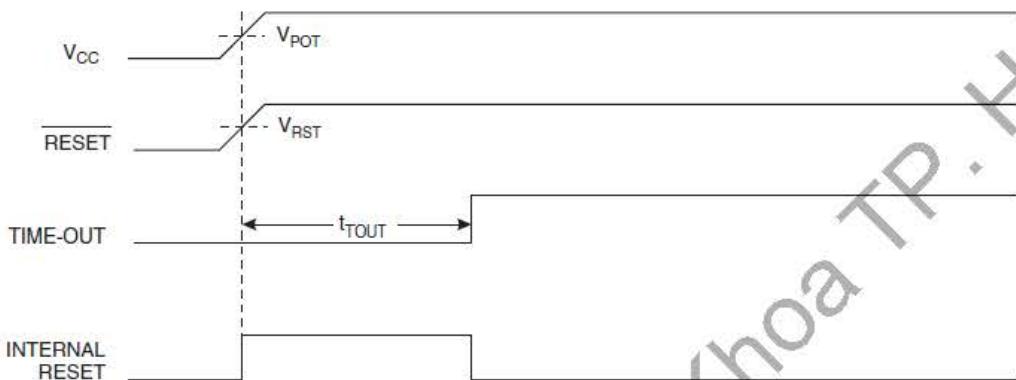
#### 2.8.1.1 Reset bật nguồn (Power-on Reset)

Xung Power-on Reset (POR) được tạo ra bởi một mạch phát hiện trên chip. POR được kích hoạt bắt đầu khi nào Vcc ở dưới mức điện áp ngưỡng VPOT (Power-on Reset Threshold) (xem bảng 2.11 phía dưới). Mạch POR có thể được sử dụng để kích hoạt reset khởi động, cũng như để phát hiện sự cố về điện áp cung cấp.



Hình 2.30: Chân RESET nối nguồn Vcc

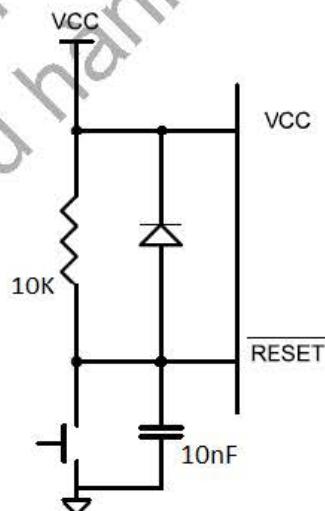
Mạch POR đảm bảo rằng thiết bị được reset khi bật nguồn. Khi đạt đến điện áp ngưỡng  $V_{POT}$ , sẽ kích hoạt bộ đếm tạo trễ. bộ đếm này xác định khoảng thời gian thiết bị được giữ thêm trạng thái reset sau khi  $V_{CC}$  tăng. Khi hết thời gian này thì MCU hoạt động bình thường. Khi  $V_{CC}$  lại giảm xuống dưới mức phát hiện, tín hiệu reset lập tức được kích hoạt trở lại mà không có thêm bất kỳ thời gian trễ nào.



Hình 2.31: MCU khởi động, chân RESET nối nguồn  $V_{CC}$

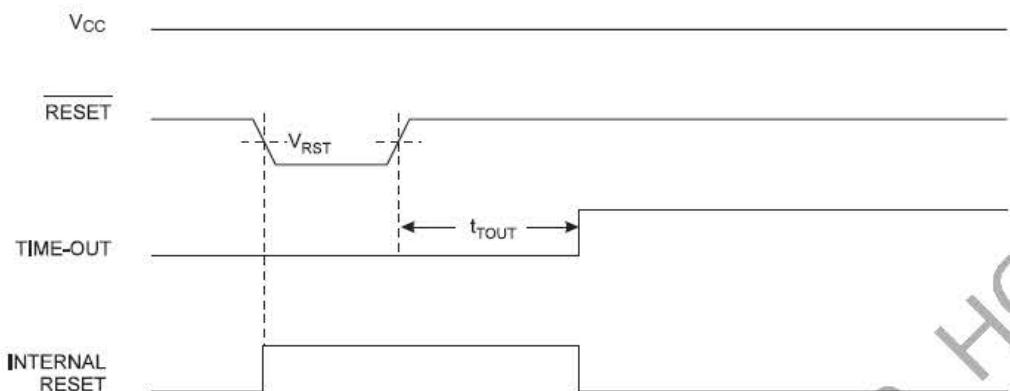
### 2.8.1.2 Reset ngoài (External Reset)

Mạch reset bên ngoài tạo ra tín hiệu tích cực thấp (mức 0) đưa vào chân RESET của MCU. Xung reset phải dài hơn độ rộng xung tối thiểu (xem bảng 2.11) sẽ tạo ra reset, ngay cả khi không có xung nhịp. Những xung ngắn hơn không đảm bảo để tạo ra reset.



Hình 2.32: Mạch reset ngoài

Khi tín hiệu reset đến điện áp ngưỡng  $V_{RST}$  trên cạnh dương của nó, sẽ kích hoạt bộ đếm tạo trễ tạo thêm khoảng thời gian reset. Khi hết thời gian này thì MCU hoạt động bình thường.

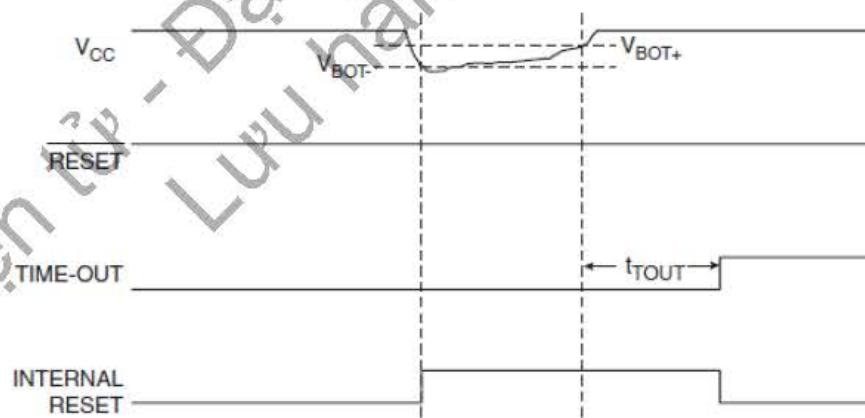


Hình 2.33: Tín hiệu reset ngoài

#### 2.8.1.3 Reset Brown-out (Brown-out Reset)

ATmega324P có mạch phát hiện sụt áp BOD, có nhiệm vụ giám sát điện áp nguồn V<sub>CC</sub> trong lúc hệ thống hoạt động bằng cách so sánh với mức điện áp ngưỡng V<sub>BOT</sub> (Brown-out Reset Threshold). V<sub>BOT</sub> tùy theo cài đặt BODLEVEL (xem bảng 2.18 ở mục 2.9.3). Mức điện áp ngưỡng có độ trễ để tránh BOD thay đổi đột ngột. Độ trễ trên BOD có thể hiểu là V<sub>BOT+</sub> = V<sub>BOT</sub> + V<sub>HYST</sub>/2 và V<sub>BOT-</sub> = V<sub>BOT</sub> - V<sub>HYST</sub>/2.

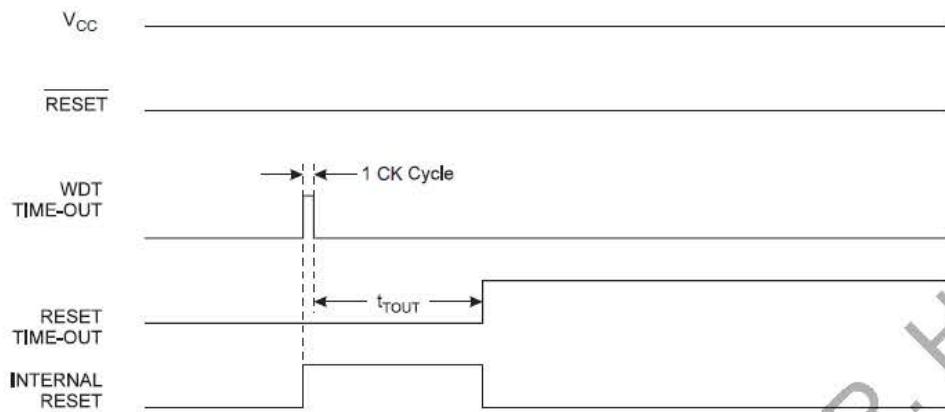
Khi BOD được kích hoạt và V<sub>CC</sub> giảm xuống giá trị dưới mức V<sub>BOT-</sub>, reset Brown-out ngay lập tức được kích hoạt. Sau đó khi V<sub>CC</sub> tăng trên mức V<sub>BOT+</sub>, sẽ kích hoạt bộ đếm tạo trễ tạo thêm khoảng thời gian reset. Khi hết thời gian t<sub>TOUT</sub> thì MCU hoạt động bình thường. Mạch BOD sẽ chỉ phát hiện sự sụt giảm trong V<sub>CC</sub> nếu điện áp ở dưới mức kích hoạt dài hơn t<sub>BOD</sub> (xem bảng 2.11).



Hình 2.34: Tín hiệu Brown-out Reset

#### 2.8.1.4 Reset hệ thống Watchdog (Watchdog System Reset)

Bộ định thời Watchdog có chức năng dùng để reset MCU trong trường hợp chương trình đang chạy bị lỗi hay bị lặp vòng vô hạn. Khi bộ định thời Watchdog được cho phép, nếu hết thời gian do bộ định thời tạo ra, nó sẽ tạo 1 xung reset có chiều dài bằng 1 chu kỳ xung nhịp. Tại thời điểm cạnh xuống của xung này sẽ kích hoạt bộ đếm tạo trễ tạo thêm khoảng thời gian reset. Khi hết thời gian t<sub>TOUT</sub> này thì MCU hoạt động bình thường.



Hình 2.35: Tín hiệu Watchdog System Reset

Hoạt động của bộ định thời Watchdog sẽ được trình bày ở phần cuối của chương này.

Bảng 2.18 Đặc tính reset

Ký hiệu	Thông số	Min	Typ	Max	Đơn vị
$V_{POT}$	Power-on Reset Threshold Voltage (rising)	0.7	1.0	1.4	V
	Power-on Reset Threshold Voltage (falling)	0.6	0.9	1.3	
$V_{RST}$	RESET Pin Threshold Voltage	$0.2V_{CC}$		$0.9V_{CC}$	
$t_{RST}$	Minimum pulse width on RESET Pin			2.5	$\mu s$
$V_{HYST}$	Brown-out Detector Hysteresis		50		mV
$t_{BOD}$	Min Pulse Width on Brown-out Reset		2		$\mu s$

#### 2.8.1.5 Reset JTAG (JTAG AVR Reset)

MCU được reset miễn là có logic 1 trong thanh ghi reset, một trong các chuỗi quét của hệ thống JTAG. Tham khảo chi tiết trong data sheet của ATmega324P.

#### 2.8.2 Thanh ghi trạng thái MCU – MCUSR (MCU Status Register)

Thanh ghi MCUSR cung cấp thông tin về các nguồn reset MCU.

Bit	7	6	5	4	3	2	1	0	MCUSR
0x34 (0x54)	-	-	-	JTRF	WDRF	BORF	EXTRF	PORF	
Read/Write				R/W	R/W	R/W	R/W	R/W	
Initial Value				0	0	0	0	0	

Hình 2.36: Thanh ghi MCUSR thông báo các nguồn reset

- Bit 4 – JTRF: cờ Reset JTAG (JTAG Reset Flag)

Bit này được đặt nếu một Reset đang diễn ra bởi logic 1 trong thanh ghi reset JTAG được chọn bởi lệnh JTAG AVR\_RESET. Bit này được xóa khi có Reset bật nguồn (Power-on Reset) hoặc bằng cách ghi logic 0 vào cờ.

- Bit 3 – WDRF: cờ Reset hệ thống Watchdog (Watchdog System Reset Flag)

Bit này được đặt nếu xảy ra Reset hệ thống Watchdog. Bit được xóa bằng Reset bật nguồn (Power-on Reset) hoặc bằng cách ghi logic 0 vào cờ.

- Bit 2 – BORF: cờ Reset Brown-out (Brown-out Reset Flag)

Bit này được đặt nếu xảy ra Reset Brown-out. Bit được xóa bằng Reset bật nguồn Power-on hoặc bằng cách ghi logic 0 vào cờ.

- Bit 1 – EXTRF: cờ Reset ngoài (External Reset Flag)

Bit này được đặt nếu xảy ra Reset ngoài. Bit được xóa bằng Reset bật nguồn Power-on hoặc bằng cách ghi logic 0 vào cờ.

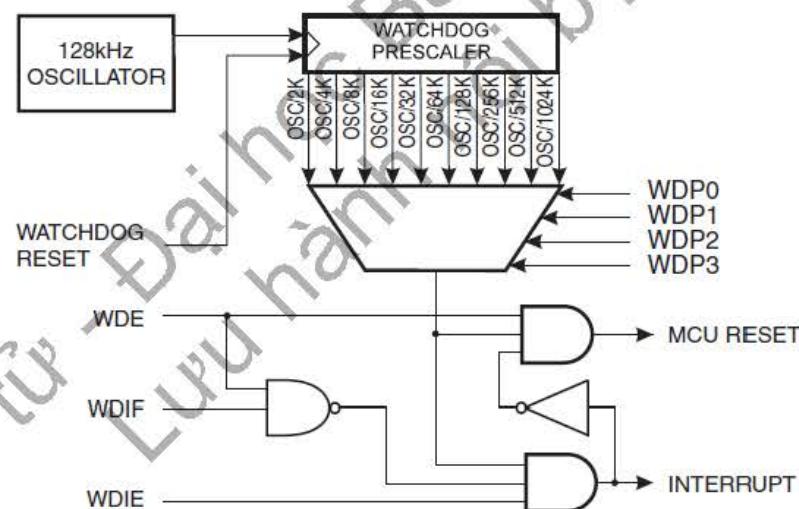
- Bit 0 – PORF: cờ Reset bật nguồn Power-on (Power-on Reset Flag)

Bit này được đặt nếu xảy ra Reset bật nguồn Power-on. Bit được xóa chỉ bằng cách ghi giá trị 0 vào nó.

### 2.8.3 Bộ định thời Watchdog (Watchdog Timer)

#### 2.8.3.1 Hoạt động bộ định thời Watchdog

Bộ định thời Watchdog (WDT) là một bộ đếm chu kỳ của một bộ dao động 128kHz riêng biệt trên chip. WDT đưa ra một ngắt hoặc reset hệ thống khi bộ đếm đạt đến giá trị thời gian chờ (time-out) nhất định. Trong chế độ hoạt động bình thường, yêu cầu hệ thống sử dụng lệnh Watchdog Timer Reset (WDR) để khởi động lại bộ đếm trước khi hết thời gian chờ time-out. Nếu hệ thống không khởi động lại bộ đếm, ngắt hoặc reset hệ thống sẽ được thực hiện.



Hình 2.37: Bộ định thời Watchdog

WDT có 3 chế độ hoạt động:

- Chế độ ngắt: WDT đưa ra một ngắt khi bộ định thời kết thúc. Ngắt này có thể được sử dụng để đánh thức thiết bị ở các chế độ ngủ và cũng như một bộ định thời hệ thống chung.
- Chế độ reset hệ thống: WDT reset khi bộ định thời kết thúc. Việc này thường được sử dụng để ngăn hệ thống treo khi đang chạy chương trình.
- Chế độ ngắt và reset hệ thống: kết hợp hai chế độ trên bằng cách đầu tiên đưa ra một ngắt và sau đó chuyển sang chế độ reset hệ thống. Chế độ này sẽ ngay lập tức cho phép tắt máy an toàn bằng cách lưu các thông số quan trọng trước khi reset hệ thống.

### 2.8.3.2 Thanh ghi điều khiển bộ định thời Watchdog – WDTCSR (Watchdog Timer Control Register)

Bit	7	6	5	4	3	2	1	0	
(0x60)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value		0	0	0		0	0	0	

Hình 2.38: Thanh ghi WDTCSR điều khiển hoạt động WDT

- Bit 7 – WDIF: cờ ngắt Watchdog (Watchdog Interrupt Flag)

Bit này được đặt khi hết thời gian chờ  $t_{TOUT}$  trong bộ định thời Watchdog (được cấu hình chế độ ngắt). Bit WDIF bị xóa bởi phần cứng khi thực thi ngắt tương ứng. Ngoài ra, WDIF có thể được xóa bằng cách ghi giá trị 1 vào nó.

- Bit 6 – WDIE: cho phép ngắt Watchdog (Watchdog Interrupt Enable)

Khi bit này được ghi giá trị 1 và bit I trong SREG được đặt, thì ngắt Watchdog được cho phép. Nếu kết hợp bit WDE=0, bộ định thời Watchdog ở chế độ ngắt. Khi hết thời gian chờ  $t_{TOUT}$  thì ngắt được thực thi. Còn nếu kết hợp bit WDE=1 thì bộ định thời Watchdog ở chế độ ngắt và reset hệ thống. Trước tiên khi hết thời gian chờ  $t_{TOUT}$  thì đặt bit WDIF. Việc thực thi ngắt sẽ xóa WDIE và cờ WDIF tự động bằng phần cứng. Bộ định thời chuyển sang chế độ reset hệ thống.

Bảng 2.19: Cấu hình bộ định thời Watchdog

WDTON	WDE	WDIE	Chế độ	Hoạt động Time-out
0	0	0	Dừng (Stopped)	Không có
0	0	1	Chế độ ngắt (Interrupt Mode)	Ngắt (Interrupt)
0	1	0	Chế độ reset hệ thống	Khởi động (Reset)
0	1	1	Chế độ reset hệ thống và ngắt	Ngắt, sau đó reset hệ thống
1	x	x	Chế độ reset hệ thống	Khởi động (Reset)

WDTON là cầu chì (xem mục 2.9.2)

- Bit 5 – WDP3: tỉ lệ Watchdog 3 (Watchdog Prescale 3)

Kết hợp với bit 0,1,2 tạo thành hệ số tỉ lệ Watchdog.

- Bit 4 – WDCE: cho phép thay đổi Watchdog (Watchdog Change Enable)

Bit này được sử dụng trong trình tự thời gian để thay đổi bit WDE và các bit tỉ lệ. Để xóa bit WDE và / hoặc thay đổi các bit tỉ lệ, phải đặt bit WDCE. Sau khi được ghi giá trị 1, phần cứng sẽ xóa WDCE sau bốn chu kỳ xung nhịp.

- Bit 3 – WDE: cho phép reset hệ thống Watchdog (Watchdog Enable)

Bit WDE bị cờ WDRF trong MCUSR ghi đè. Điều này có nghĩa là WDE luôn được đặt khi WDRF được đặt. Để xóa bit WDE, trước tiên phải xóa cờ WDRF.

- Bit 2:0 – WDP2, WDP1, WDP0: các bit tỉ lệ Watchdog (Watchdog Prescale 2, 1, 0)

Kết hợp với WDP3 (bit 5) tạo ra 4 bit tỉ lệ của bộ định thời Watchdog. Mỗi tổ hợp tương ứng với khoảng thời gian chờ của bộ định thời Watchdog.

**Bảng 2.20:** Lựa chọn tỉ lệ bộ định thời Watchdog

WDP3	WDP2	WDP1	WDP0	Số chu kỳ dao động WDT	Thời gian Time-out (Vcc = 5.0V)
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32768) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s
1	0	1	0	Reserved	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

## 2.9 CÁC BIT CẦU CHÌ (FUSE BITS)

Có một số tính năng của AVR mà chúng ta có thể chọn bằng cách lập trình các bit cầu chì (fuse bits). Các tính năng này giúp giảm chi phí hệ thống bằng cách bỏ các nhu cầu đối với các thành phần bên ngoài.

Các bit cầu chì này là phần cứng không được thiết lập bằng lệnh trong chương trình. Nó được thiết lập riêng với file chương trình. Mỗi phần mềm nạp chương trình vào chip đều có hỗ trợ phần thiết lập cho các bit cầu chì này.

Đối với các bit cầu chì, gán giá trị 0 cho bit cầu chì đồng nghĩa với bit đó đã được lập trình (programmed), trong khi gán giá trị 1 nghĩa là chưa được lập trình (unprogrammed).

ATmega324P có 3 byte chứa các bit cầu chì. Mỗi bit có 1 giá trị mặc định ban đầu là 0 (programmed) hoặc 1 (unprogrammed).

### 2.9.1 Byte thấp (Fuse Low Byte)

Bảng 2.21: Byte thấp cầu chì

Byte thấp cầu chì (Fuse Low Byte)	Bit	Mô tả	Giá trị mặc định
<b>CKDIV8</b>	7	Chia 8 xung nhịp (Divide clock by 8)	<b>0</b> (đã lập trình)
<b>CKOUT</b>	6	Ngõ ra xung nhịp (Clock output)	<b>1</b> (chưa lập trình)
<b>SUT1</b>	5	2 bit chọn thời gian khởi động	<b>1</b> (chưa lập trình)
<b>SUT0</b>	4	(Select start-up time)	<b>0</b> (đã lập trình)
<b>CKSEL3</b>	3		<b>0</b> (đã lập trình)
<b>CKSEL2</b>	2	4 bit chọn nguồn xung nhịp (Select Clock Source)	<b>0</b> (đã lập trình)
<b>CKSEL1</b>	1		<b>1</b> (chưa lập trình)
<b>CKSEL0</b>	0		<b>0</b> (đã lập trình)

- Bit 7 – **CKDIV8**: chia 8 xung nhịp  
Bit có giá trị mặc định là 0, nghĩa là tần số xung nhịp kích MCU làm việc=1/8 tần số xung nhịp chính (dao động nội hay ngoại)
- Bit 6 – **CKOUT**: ngõ ra xung nhịp  
Khi bit này được lập trình thì cho phép nguồn xung nhịp hệ thống xuất ra chân port PORTB1. Giá trị mặc định là 1 nghĩa là chưa cho phép.
- Bit 5:4 – **SUT1:SUT0**: chọn thời gian khởi động (Start-Up Time)  
2 bit SUT1, SUT0 và bit CKSEL0 dùng chọn thời gian khởi động. Đọc thêm data sheet để hiểu đầy đủ chi tiết.
- Bit 3:0 – **CKSEL3, CKSEL2, CKSEL1, CSKEL0**: Chọn nguồn xung nhịp  
4 bit này dùng để chọn nguồn xung nhịp cho hệ thống (theo bảng 2.8)  
Giá trị mặc định của CKSEL3..0 là 0010, nguồn xung clock mặc định được chọn là dao động RC nội 8MHz. Với cầu chì CKDIV8 cũng được cho phép, thì xung nhịp hệ thống mặc định sẽ là 1MHz.

### 2.9.2 Byte cao (Fuse High Byte)

Bảng 2.22 : Byte cao cầu chì

Byte cao cầu chì (Fuse High Byte)	Bit	Mô tả	Giá trị mặc định
<b>OCDEN</b>	7	Cho phép OCD (Enable OCD)	<b>1</b> (chưa lập trình, cấm OCD)
<b>JTAGEN</b>	6	Cho phép JTAG (Enable JTAG)	<b>0</b> (đã lập trình, cho phép JTAG)
<b>SPIEN</b>	5	Cho phép tải chương trình và dữ liệu nối tiếp (Enable Serial Program and Data Downloading)	<b>0</b> (đã lập trình, cho phép lập trình SPI)
<b>WDTON</b>	4	Bộ định thời Watchdog luôn mở (Watchdog Timer always on)	<b>1</b> (chưa lập trình)

<b>EESAVE</b>	3	EEPROM được bảo toàn khi xóa chip (EEPROM memory is preserved through the Chip Erase)	<b>1</b> (chưa lập trình, không bảo toàn EEPROM) <b>0</b> (đã lập trình)
<b>BOOTSZ1</b>	2	2 bit chọn kích thước vùng khởi động (Select Boot Size)	<b>0</b> (đã lập trình) <b>1</b> (chưa lập trình)
<b>BOOTSZ0</b>	1		
<b>BOOTRST</b>	0	Chọn vec tơ khởi động (Select Reset Vector)	<b>1</b> (chưa lập trình)

- Bit 7 – **OCDEN**: cho phép gỡ rối trên chip (On-Chip Debug Enable)
- Bit 6 – **JTAGEN**: cho phép giao tiếp JTAG (JTAG ENable)
- Bit 5 – **SPIEN**: cho phép tải chương trình và dữ liệu nối tiếp (Serial Port Interface ENable)
- Bit 4 – **WDTON**: bộ định thời Watchdog luôn mở (WatchDog Timer On)
- Bit 3 – **EESAVE**: EEPROM được bảo toàn khi xóa chip (EEprom SAVE)
- Bit 2:1 – **BOOTSZ1:BOOTSZ0**: chọn kích thước vùng khởi động (BOOT SiZe)

**Bảng 2.23:** Kích thước vùng khởi động

BOOTSZ1	BOOTSZ0	Kích thước vùng khởi động	Số trang	Vùng Flash ứng dụng	Vùng Flash khởi động	Địa chỉ cuối vùng ứng dụng	Địa chỉ đầu vùng khởi động
1	1	128 words	2	0x0000 - 0x1F7F	0x1F80 - 0xFFFF	0x1F7F	0x1F80
1	0	256 words	4	0x0000 - 0x1EFF	0x1F00 - 0xFFFF	0x1EFF	0x1F00
0	1	512 words	8	0x0000 - 0x1DFF	0x1E00 - 0xFFFF	0x1DFF	0x1E00
0	0	1024 words	16	0x0000 - 0x1BFF	0x1C00 - 0xFFFF	0x1BFF	0x1C00

Giá trị đầu của 2 bit này là 00, như vậy vùng khởi động có kích thước lớn nhất là 1024 word. Vùng khởi động có địa chỉ từ 0x1C00 – 0xFFFF.

- Bit 0 – **BOOTRST**: chọn vec tơ khởi động (BOOT ReSeT)  
Bit này là 1 (chưa lập trình) sẽ chọn vec tơ khởi động là địa chỉ đầu của vùng Flash ứng dụng là 0x0000. Ngược lại nếu bit này là 0 (đã lập trình) thì vec tơ khởi động là địa chỉ đầu của vùng Flash khởi động (xem bảng 2.23)

### 2.9.3 Byte mở rộng (Fuse Extended Byte)

**Bảng 2.24:** Byte mở rộng cầu chì

Byte mở rộng cầu chì (Fuse Extended Byte)	Bit	Mô tả	Giá trị mặc định
-	7	-	<b>1</b>
-	6	-	<b>1</b>

-	5	-	1
-	4	-	1
-	3	-	1
<b>BODLEVEL2</b>	2	3 bit chọn mức kích bộ BOD	1 (chưa lập trình)
<b>BODLEVEL1</b>	1	(Brown-out Detector trigger level)	1 (chưa lập trình)
<b>BODLEVEL0</b>	0		1 (chưa lập trình)

- Bit 2:0 – **BODLEVEL2, BODLEVEL1, BODLEVEL0**: mức kích bộ BOD (BOD LEVEL)

Các bit này cho phép chọn lựa mức điện áp  $V_{BOT}$  khác nhau như bảng 2.25. Giá trị đầu của 3 bit này là 111, nghĩa là cấm bộ BOD.

**Bảng 2.25:** Cầu chì BOD

BODLEVEL 2:0	Min $V_{BOT}$	Typ $V_{BOT}$	Max $V_{BOT}$	Đơn vị	
111	BOD Disabled				
110	1.7	1.8	2.0	V	
101	2.5	2.7	2.9		
100	4.1	4.3	4.5		
011	Dự trữ				
010	Dự trữ				
001	Dự trữ				
000	Dự trữ				

Trong phần này chỉ khảo sát tính năng một số cầu chì cơ bản, cần đọc thêm data sheet của ATmega324P để biết thêm các cầu chì khác. Cách thiết lập các cầu chì tùy theo chương trình hỗ trợ lập trình.

## BÀI TẬP

- Thanh ghi PC có bao nhiêu bit? Tại sao?
- ALU dùng trên các toán hạng bao nhiêu bit?
- Tính chất cơ bản của kỹ thuật đường ống (pipeline) là gì?
- Vai trò của các thanh ghi I/O là gì?
- Thanh ghi I/O chuẩn và thanh ghi I/O mở rộng có điểm khác nhau cơ bản là gì?
- Cho biết sự khác nhau của cờ N và cờ S? Cho ví dụ
- Nếu R16 chứa giá trị \$5D, để có cờ Z =1 sau khi thực hiện lệnh ADD R16, R17 thì nội dung của thanh ghi R17 phải là bao nhiêu?
- Dùng lệnh ADD, viết đoạn chương trình thực hiện phép toán nhân 10 lần giá trị thanh ghi R16. Giả sử kết quả là 8 bit, cắt kết quả vào thanh ghi R17.
- Thanh ghi R16 chứa giá trị \$2A, thanh ghi R17 chứa giá trị \$F0. Hãy cho biết khi thực hiện đoạn chương trình sau, thì nội dung 2 thanh ghi trên thay đổi như thế nào?

PUSH R16

PUSH R17

POP R16

POP R17

- Viết đoạn chương trình nhập dữ liệu từ Port B và xuất ra Port C.

- Trong 3 thanh ghi GPIO<sub>Rn</sub> (n= 0,1,2), thanh ghi nào cho phép truy xuất địa chỉ bit trực tiếp

12. Thanh ghi EEAR của ATmega324P sử dụng bao nhiêu bit? Vì sao?
13. Thanh ghi nào là nội dung của bộ đếm Timer0? Có địa chỉ bộ nhớ và địa chỉ I/O (nếu có) là bao nhiêu?
14. Trong các thanh ghi cổng nối tiếp, thanh ghi nào có địa chỉ I/O? địa chỉ bộ nhớ của chúng?
15. Thanh ghi nào chứa nội dung của bộ ADC? Có địa chỉ bộ nhớ và địa chỉ I/O (nếu có) là bao nhiêu?
16. Trong các thanh ghi ngắn, thanh ghi nào có địa chỉ I/O? địa chỉ bộ nhớ của chúng?
17. Chế độ ngủ (sleep mode) là gì? Vì sao cần có nhiều chế độ khác nhau?
18. Sau khi reset, thanh ghi PC có giá trị là bao nhiêu? Các cổng I/O là cổng nhập hay cổng xuất?
19. ATmega 324P có bao nhiêu nguồn reset?
20. Bit cầu chì (Fuse Bit) là gì? ATmega 324P có bao nhiêu byte cầu chì?

Bộ môn Điện tử - Đại học Bách Khoa TP. HCM  
Lưu hành nội bộ