

Lời mở đầu

Ngày nay, với sự phát triển nhanh chóng của khoa học kỹ thuật, nó đã dần dần giảm bớt, thay thế sức lao động của con người trong mọi mặt của đời sống. Đặc biệt là trong ngành sản xuất chế biến thức ăn chăn nuôi, vốn đã và đang đem lại nguồn thực phẩm dồi dào cho con người. Quá trình ứng dụng đó đòi hỏi phải có khả năng tự động hóa cao, đáp ứng được các yêu cầu về công nghệ.

Vi điều khiển ngày nay được ứng dụng rộng rãi trong các dây chuyền tự động hóa, với với sự vượt trội về khả năng đáp ứng nhanh, giá thành rẻ, chất lượng tín hiệu tin cậy... nó đã thực sự trở thành một nền tảng để nghiên cứu và ứng dụng vào các dây chuyền sản xuất.

Với tầm quan trọng và sự phát triển của công nghiệp chế biến, sản xuất thức ăn chăn nuôi nên em đã nhận đề tài “ **Trang bị điện máy sản xuất thức ăn chăn nuôi, đi sâu thiết kế hệ thống sấy khô**” làm đề án tốt nghiệp của mình. Đề tài của em gồm có 3 chương:

Chương 1: **Giới thiệu về quy trình sản xuất thức ăn chăn nuôi**

Chương 2: **Các phần tử sử dụng trong hệ thống sấy băng tải**

Chương 3: **Thiết kế và thi công hệ thống sấy băng tải**

Trong quá trình làm đề án tốt nghiệp, do sự hạn chế về thời gian, tài liệu và trình độ có hạn nên không tránh khỏi có thiếu sót. Em rất mong được sự đúng góp ý kiến của thầy cô và các bạn để đề án tốt nghiệp của em được hoàn thiện hơn, cũng như có được kiến thức cần thiết để ra đời làm thật.

Em xin gửi lời cảm ơn chân thành đến các thầy cô trong khoa Điện tự động công nghiệp, và đặc biệt là GS.TSKH. Thân Ngọc Hoàn, người đã trực tiếp giúp đỡ em hoàn thành tốt đề án này.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Bùi Tiến Tùng

CHƯƠNG 1.

GIỚI THIỆU VỀ QUY TRÌNH SẢN XUẤT THỨC ĂN CHĂN NUÔI

1.1. GIỚI THIỆU CHUNG VỀ TÌNH HÌNH SẢN XUẤT THỨC ĂN CHĂN NUÔI

1.1.1. Sơ lược về tình hình sản xuất thức ăn viên ở nước ta.

Trong những năm 1975, ngành công nghiệp nói chung và công nghiệp chế biến nói riêng nước ta còn gặp nhiều khó khăn do ảnh hưởng nặng nề của chiến tranh nên quy trình sản xuất tập trung còn hạn chế, lạc hậu. Mặc dù là nước nông nghiệp song lúc này chưa được ưu tiên phát triển, đặc biệt là sản xuất thức ăn chăn nuôi, cho nên trên thị trường cũng như ngành chăn nuôi Việt Nam vắng mặt thức ăn chăn nuôi dạng viên, chỉ có thức ăn dạng bột mà thôi. Mãi tới đầu năm 1993 thức ăn chăn nuôi dạng viên thực sự trở lại với thị trường Việt Nam, nó được xem là sự hiện diện mới nhất và mang lại cho ngành chăn nuôi nhiều lợi ích.

Đi tiên phong là xí nghiệp thức ăn gia súc VIFOCO đã đưa thức ăn dạng viên vào quy trình sản xuất của xí nghiệp vào tháng 2 năm 1993, với nhiều thiết bị nhập từ Mỹ. Sau đó xí nghiệp đã nhập bộ khuôn mới, từ đó bắt đầu đi vào ổn định với năng suất cả nhà máy có thể đạt từ 4-6 (tấn/h). Nhưng sản phẩm xí nghiệp lúc này vẫn chưa được tiêu thụ mạnh do người nông dân chưa quen áp dụng loại thức ăn này vào trong chăn nuôi đồng thời giá thành còn cao, chất lượng còn thấp do hệ thống quá cũ.

Tiếp theo sau đó vào tháng 9 năm 1993 xí nghiệp chế biến thức ăn gia súc Việt Thái đã phục hồi dây chuyền sản xuất tương tự như dây truyền sản xuất của xí nghiệp VIFOCO, năng suất có thể đạt từ 4-6 (tấn/h) nhưng vấn đề về chất lượng thời gian đầu vẫn chưa được thỏa mãn, xong xí nghiệp đã đạt được những thành quả nhất định.

Tháng 7 năm 1994 xí nghiệp chế biến thức ăn gia súc PROCONO bắt đầu đi vào hoạt động với dây chuyền sản xuất thức ăn viên của Pháp, năng suất 6 tấn/h.

Tháng 1/1995 nhà máy chế biến thức ăn An Phú đã tiến hành lắp ráp dây chuyền ép viên và đi vào hoạt động tháng 3/1995 với dây chuyền máy Pellet.

Cho đến hiện nay, thức ăn chăn nuôi dạng viên đã được sử dụng rộng rãi ở nước ta. Nhiều nhà máy thức ăn chăn nuôi đã trang bị hệ thống sản xuất thức ăn dạng viên tiên tiến, như vào tháng 5/2005 nhà máy thức ăn gia súc Bình Minh đã lắp đặt hệ thống thức ăn chăn nuôi dạng viên của Buhler (Thụy Sĩ).

Như vậy, thức ăn gia súc dạng viên chỉ thực sự đến với ngành chăn nuôi vào đầu năm 1993. Thời kỳ đầu đã gặp không ít khó khăn trong sản xuất cũng như trong tiêu thụ. Do đa số thiết bị là phục hồi lại nên sản phẩm chưa đảm bảo yêu cầu kỹ thuật, thị trường chưa quen sử dụng thức ăn dạng viên vào chăn nuôi, giá thành còn cao. Nhưng hiện nay với trang thiết bị mới, hiện đại, chất lượng và năng suất sản phẩm được cải thiện đáng kể và ổn định được giá thành nên thức ăn dạng viên chiếm một vị trí quan trọng trong ngành chăn nuôi ở nước ta, đặc biệt là ngành chăn nuôi gia súc, gia cầm, thủy cầm.

1.1.2. Các nguồn nguyên liệu thường dùng làm thức ăn cho gia súc , gia cầm, thủy cầm (GSTC).

Nguyên liệu thức ăn là sản phẩm từ nguồn gốc động thực vật, vi sinh vật, chất khoáng và những chất tổng hợp hóa học khác. Những nguyên liệu thức ăn này vừa bảo đảm cung cấp chất dinh dưỡng cho nhu cầu sinh lý sinh trưởng phát triển sinh sản của GSTC vừa mang tính chất kích thích tăng trưởng, tăng sức khỏe chống lại bệnh và vừa dễ hấp thu. Căn cứ vào thành phần dinh dưỡng có trong nguyên liệu và hàm lượng của chúng, các nguyên liệu chính thường dùng làm thức ăn cho GSTC gồm các nhóm:

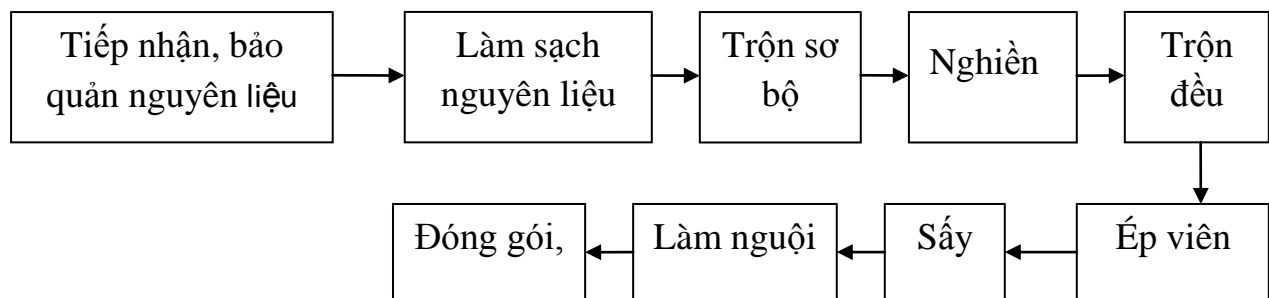
- + Thức ăn tinh bột: Ngô, thóc, cám gạo, kê, mỳ,...
- + Thức ăn giàu Protein: Đỗ tương, lạc, bột cá, bột đậu tằm,...
- + Thức ăn giàu khoáng: Gồm các phức hợp canxi photpho, muối amoni NaCl, muối của khoáng vi lượng
- + Thức ăn giàu Vitamin: hỗn hợp vitamin A, D, E, B₁, B₂, B₁₂...
- + Thức ăn bổ sung : Gồm một số loại thuốc giúp cân bằng tốt các dưỡng chất và phòng bệnh, chống nấm mốc gây bệnh, kích thích tăng trưởng : antibiotit, antihemi, anzin...

1.2. PHƯƠNG PHÁP, QUY TRÌNH SẢN XUẤT THỨC ĂN CHĂN NUÔI

1.2.1. Công nghệ sản xuất thức ăn cho GSTC

Thức ăn cho GSTC nuôi công nghiệp là loại thức ăn được hỗn hợp từ các nguồn nguyên liệu khác nhau, thông qua các công nghệ nghiền, trộn, gọi là thức ăn hỗn hợp dưới dạng bột hoặc viên. Thức ăn hỗn hợp chứa đầy đủ các vật chất dinh dưỡng cần thiết cho sinh lý phát triển và sinh sản của gia cầm đem lại hiệu quả cao như: protein, năng lượng, vitamin và chất khoáng. Ngoài ra còn được bổ sung các chất kích thích sinh trưởng như các enzym, các kháng sinh...

Thành phần các nguyên liệu dùng để phối trộn và nhu cầu thành phần dinh dưỡng cũng không khác gì so với các loại thức ăn bột, sợi. Về mặt cơ cấu nguyên liệu dùng để sản xuất thường là: bột cá, bột ngũ cốc, các loại đậu. Về thành phần dinh dưỡng chúng cũng cần chủ yếu là protein, glucit, lipit, vitamin, các khoáng chất,...



Hình 1.1: Sơ đồ công nghệ sản xuất thức ăn viên.

1.2.2. Đặc điểm và yêu cầu kích thước của thức ăn chăn nuôi.

Các viên thức ăn chăn nuôi tạo ra có thể có dạng hình trụ, lăng trụ hoặc viên định hình.

Độ dài của viên được xác định phụ thuộc vào đường kính của viên, thường tỷ lệ giữa độ dài của viên và đường kính của viên là: (1,3-1,4):1

Bảng 1 trình bày số liệu của công ty Stolz (Pháp) về đường kính của viên đối với một số vật nuôi.

Bảng 1.1: Số liệu về đường kính của viên đối với một số vật nuôi.

Loại vật nuôi	Đường kính viên (mm)
Trâu, bò	8
Heo	6
Gà, vịt	4
Tôm, cá	2,5

- độ cứng của viên: độ cứng của viên có vai trò quan trọng, nếu viên quá cứng sẽ tổn công nhai và đôi khi không tận dụng được nguồn dinh dưỡng mà chúng ta cung cấp cho chúng. Nếu không đủ độ cứng sẽ dễ bị bể vỡ trong quá trình vận chuyển. Độ cứng còn phụ thuộc vào áp suất ép, đường kính lỗ, chiều

dài lỗ khuôn, tính chất của nguyên liệu chế ra nó. Tùy theo đường kính của viên mà có độ cứng được đánh giá qua lực phá vỡ của viên như sau:

- + Đường kính của viên đến 4 mm chịu lực phá vỡ 50 N.
- + Đường kính của viên đến 8 mm chịu lực phá vỡ 60 N.
- + Đường kính của viên trên 8 mm chịu lực phá vỡ 80 N.

Viên phải có độ bền, chịu được sự rung động, viên đưa vào đóng bao phải có độ ẩm ở chế độ bảo quản (dưới 14%), và nhiệt độ bền bằng nhiệt độ môi trường. Viên cần có độ đồng đều cao. Năng suất cầu máy phải cao, chi phí năng lượng riêng phải thấp khoảng 50 kWh/tấn cho viên có đường kính $d = 2,5$ mm; 15-20 kWh/tấn cho viên có đường kính $d=(6-8)$ mm.

1.2.3. Phương pháp xác định độ nhỏ bột nghiền

Độ nhỏ bột nghiền là kích thước hình học của các phần tử bột nghiền. Đối với một thể tích khối bột người ta dùng kích thước trung bình của khối bột để đặc trưng cho độ nhỏ của bột, vì các phần tử bột nghiền có kích thước đa phân tán. Phương pháp xác định tương tự như khi đo cho các sản phẩm rời.

Phương pháp xác định độ nhỏ bột nghiền:

Căn cứ vào kích thước hạt bột mà ta có các phương pháp xác định như sau:

- Phương pháp phân tích sàng, dùng sàng để sàng thành các lớp nếu các phần tử có kích thước lớn hơn $40 \mu m$.
- Phương pháp lắng tụ: áp dụng cho các phần tử có kích thước giới hạn từ $5-10 \mu m$.
- Phương pháp soi kính hiển vi: áp dụng cho các phần tử có kích thước $< 50 \mu m$. Bằng cách đo kích thước chiều dài (tuyến tính) đặc thù của các phần tử được quan sát dưới kính hiển vi qua lưới đo của thị kính.

Thiết bị xác định thành phần kích thước hạt bằng phép phân tích sàng thường dùng loại máy sàng kiểu treo. Các phương pháp sàng đã được thống nhất hóa, các sàng sử dụng trong máy sàng kiểu treo thường là loại sàng kim loại đột lỗ, loại sợi kim loại hay loại sợi kim loại đan. Ở Liên Xô trước đây các kích thước sàng thử nghiệm với loại nhỏ được chọn theo tiêu chuẩn lỗ có kích thước $\geq 40 \mu m$. Còn lỗ sàng lớn hơn theo tiêu chuẩn có lỗ từ 1- 2,5 mm.

Để sàng sản phẩm nghiền từ các nguyên liệu thức ăn gia súc, người ta sử dụng sàng đột lỗ với kích thước lỗ hình tròn khi kích thước các phần tử ≥ 1 mm. Nếu kích thước < 1 mm thì dùng sàng bằng sợi đan lỗ vuông. Các lỗ sàng được bố trí trong một hộp lần lượt từ lỗ to đến lỗ nhỏ kể từ trên xuống, và dưới cùng là tấm đáy không khoan lỗ. Trong ngành công nghiệp chế biến thức ăn chăn nuôi, độ nhỏ hạt được xác định theo tiêu chuẩn TCVN 1535-93. Theo tiêu chuẩn này, đường kính trung bình của các phần tử được xác định theo công thức sau:

$$M = \frac{(0,5P_0 + 1,5P_1 + 2,5P_2 + 3,5P_3)}{100} (mm) \quad (1.1)$$

Trong đó:

+ P_0 : Tỷ lệ phần tử có trên đáy sàng, (%);

+ P_1, P_2, P_3 : Tỷ lệ các phần tử có trên các mặt sàng tương ứng với các đường kính D_1, D_2, D_3 (%).

Mẫu phân tích có khối lượng 100g được sàng qua các bộ sàng đập với kích thước lỗ $D = 5, 3, 2, 1$ mm khi nghiền thô và nghiền trung bình, còn $D = 4, 3, 2, 1$ và $0,2$ mm khi nghiền nhỏ. Các sàng trên cùng với lỗ $D = 5$ và 4 mm là các sàng dùng để kiểm tra và tính toán các hạt nguyên có trong mẫu. Sự có mặt của các hạt nguyên này chứng sản phẩm không đạt yêu cầu.

1.2.4. Phương pháp xác định độ trộn đều bột nghiền.

Trộn là quá trình kết hợp các khối lượng của các vật liệu khác nhau với mục đích nhận được một hỗn hợp đồng nhất của các phần tử ở mỗi cấu tử trong tất cả khối lượng hỗn hợp, bằng cách sắp xếp lại chúng dưới tác dụng của ngoại lực. Hỗn hợp tạo ra như thế để tăng cường quá trình trao đổi nhiệt và trao đổi khối lượng.

Ta có thể sử dụng phương pháp KHAPHARROP để xác định độ trộn đều của bột nghiền:

Có thể xác định độ nhỏ bột nghiền dựa vào tỷ số giữa tỷ lệ chứa của mỗi thành phần trong từng mẫu đo C_i với tỷ lệ chứa của thành phần có trong hỗn hợp C_o . Thành phần được chọn kiểm tra (mẫu kiểm tra) là thành phần có tỷ lệ nhỏ nhất trong hỗn hợp.

Sau khi lấy ra n mẫu đo và xác định tỷ lệ C_i trong từng mẫu ta tính độ trộn đều K với 2 trường hợp: $C_i \leq C_o$ và $C_i \geq C_o$.

Nếu $C_i \leq C_o$ thì:

$$K_1 = \frac{\sum_{i=1}^{n_1} \frac{C_i}{C_o}}{n_1} \quad (1.2)$$

Nếu $C_i \geq C_o$ thì:

$$K_1 = \frac{\sum_{i=1}^{n_1} \frac{100 - C_i}{C_o}}{n_1} \quad (1.3)$$

Độ trộn đều K là giá trị trung bình cộng của hai lớp mẫu đo n_1 và n_2 :

$$K = (n_1 \cdot K_1 + n_2 \cdot K_2) / (n_1 + n_2) \quad (1.4)$$

Trong đó : C_i – Tỷ lệ thành phần kiểm tra có trong mẫu thứ i ;

C_o – Tỷ lệ thành phần kiểm tra có trong toàn bộ hỗn hợp;

n_1 – Số mẫu có tỉ lệ thành phần tra $C_i \leq C_o$

n_2 – Số mẫu có tỉ lệ thành phần tra $C_i \geq C_o$

1.2.5. Phương pháp xác định độ bền và độ cứng viên thức ăn

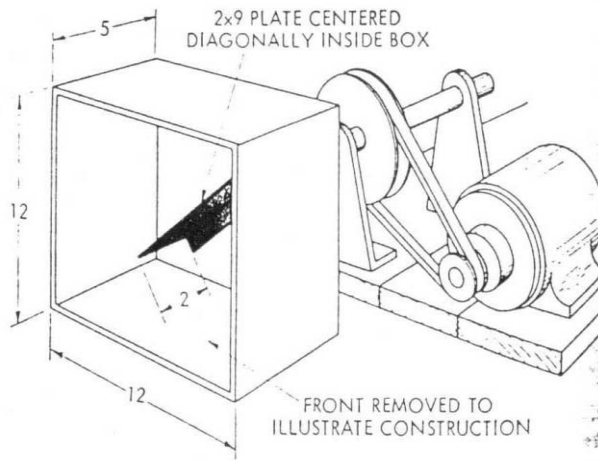
Độ bền viên thức ăn là khả năng thắng được sự tác động của ngoại lực hoặc môi trường của viên thức ăn mà nó vẫn giữ được hình dạng ban đầu và không bị phá hủy.

Có thể xác định độ bền viên thức ăn bằng phương pháp chuyên dùng hay phương pháp ngâm nước.

+ Phương pháp chuyên dùng:

Độ bền viên thức ăn là tỉ lệ viên thức ăn không bị phá hủy sau khi chịu tác động cơ học trong một thiết bị đo dùng là sàng lưới hay máy đảo trộn có gắn cánh trộn (hình 1.2). Thiết bị là hộp chữ nhật kín, có nắp mở ở phía trên có kích thước (12 x 5 x 12) in. Phía trong hộp có đặt một tấm phẳng kích thước (2 x 9) in truyền động quay cho tấm phẳng bằng động cơ điện. Cách đo như sau:

Cho 500gam thức viên thức ăn cần kiểm tra độ bền vào hộp, đóng nắp lại cho quay trong thời gian 10 phút. Sau đó lấy ra và tiến hành sàng để loại các thành phần có kích thước nhỏ.



Hình 1.2: Thiết bị kiểm tra độ bền viên thức ăn.

Độ bền viên thức ăn được xác định theo công thức:

$$D_b = \frac{W_1}{W_2} 100(\%) \quad (1.5)$$

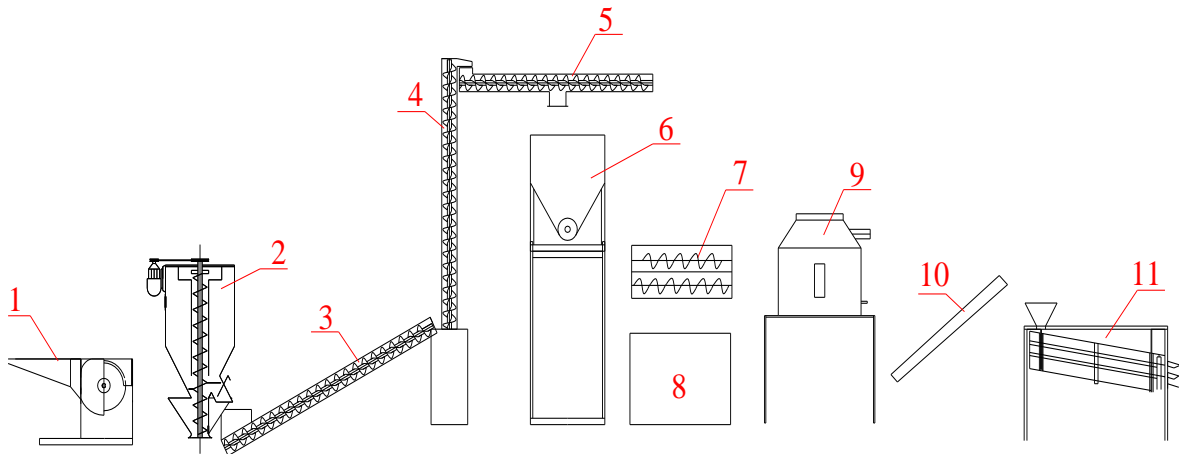
Trong đó: W_1 – Khối lượng của thức ăn nằm trên sàng sau khi rây, gam;

W_2 – Khối lượng của viên thức ăn sau khi rây, 500g.

Ngoài ra, có thể dùng phương pháp ngâm nước để đo độ bền của viên như sau:

Độ bền viên thức ăn được đặc trưng bởi thời gian ngâm nước mà viên thức ăn không bị phá hủy. nước cho vào cốc chiếm 2/3 thể tích cốc (1000 ml), cho 100 gam viên thức ăn đã có nước. Quan sát và bấm thời gian bắt đầu từ khi bỏ thức ăn vào cho tới khi bắt đầu tan.

1.2.6. Sơ bộ các công đoạn sản xuất thức ăn chăn nuôi



Hình 1.3: Sơ đồ công nghệ dây chuyền sản xuất thức ăn viên chăn nuôi.

1. Máy nghiền; 2. Máy trộn vít đứng; 3. Vít tải đứng; 4. Bun ke và Vít tải ngang;
5. Máy ép viên; 6. Băng tải; 7. Máy sấy và làm nguội; 8. Nồi hơi; 9. Quạt ly tâm; 10. Xyclon lọc bụi; 11. Sàng phân loại.

Nguyên lý làm việc:

Nguyên liệu thô chưa đạt độ nhỏ cần thiết được nghiền nhỏ bằng máy nghiền (1). Sản phẩm nghiền được vô bao để thuận tiện cho việc cân định lượng và nạp liệu vào máy trộn. Các thành phần được định lượng bằng cân thủ công và nạp trực tiếp vào trong máy trộn. Sau khi trộn xong, sản phẩm thu được là thức ăn hỗn hợp chăn nuôi dạng bột.

Để tạo hình viên thức ăn, thức ăn hỗn hợp dạng bột được ép viên bằng máy ép viên kiểu cối vòng con lăn. Thức ăn hỗn hợp dạng bột được nạp vào bun ke 4 bằng vít tải đứng 3. Đáy bun ke 4 có bố trí vít tải ngang và có gắn các cánh nạp liệu để cung cấp liên tục hỗn hợp vào máy ép viên 5. Trước khi đưa vào ép bằng cối vòng – con lăn, hỗn hợp được gia ẩm và làm chín. Sản phẩm ra khỏi máy ép viên có độ ẩm từ 26 – 18 % và nhiệt độ từ 55 – 65⁰C. Băng tải nghiêng 6 sẽ vận chuyển chúng vào buồng sấy để làm khô đến độ ẩm yêu cầu. Phía dưới

buồng sấy là buồng làm nguội bằng không khí bên ngoài.

Sản phẩm thoát khỏi buồng làm nguội nhờ cơ cấu gạt kiểu culit nằm phía dưới buồng làm nguội để rơi vào máy sàng lắc phẳng. Sàng lắc phẳng 10 phân sản phẩm ép viên đã được làm khô và thổi nguội thành 3 loại: Loại lớn, loại đạt yêu cầu và loại nhỏ. Loại lớn được đưa đi làm nhỏ bằng máy nghiền, để cùng với sản phẩm loại nhỏ đưa trở về ép viên lại. Sản phẩm đạt yêu cầu được vô bao để chuyên giao, sử dụng hay lưu kho.

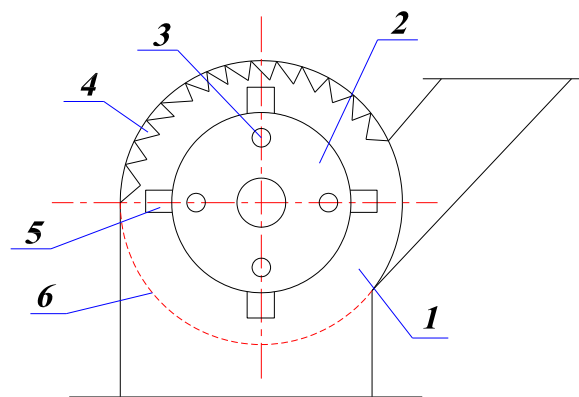
1.2.6.1. Công đoạn nghiền

Quá trình phân chia một vật thể rắn ra thành những vật thể nhỏ hơn dưới tác động của các lực ở bên ngoài được gọi là nghiền. Mức độ nghiền được đặc trưng bởi tỷ lệ giữa kích thước các tiểu phần của nguyên liệu trước khi nghiền (d_t) và kích thước các tiểu phần sau khi nghiền (d_s):

$$i = \frac{d_t}{d_s} \quad (1.6)$$

Nhiệm vụ: Làm nhỏ nguyên liệu đến kích thước yêu cầu.

Cấu tạo máy nghiền kiểu búa:



Hình 1.4: Cấu tạo máy nghiền.

Thân máy; 2.Rô to; 3. Chốt treo búa; 4. Má đập phụ; 5. Búa nghiền; 6.Sàng.

1.2.6.2. Công đoạn định lượng

Nhiệm vụ: Định lượng các thành phần (cấu tử) theo thực đơn (công thức) quy định của kỹ thuật.

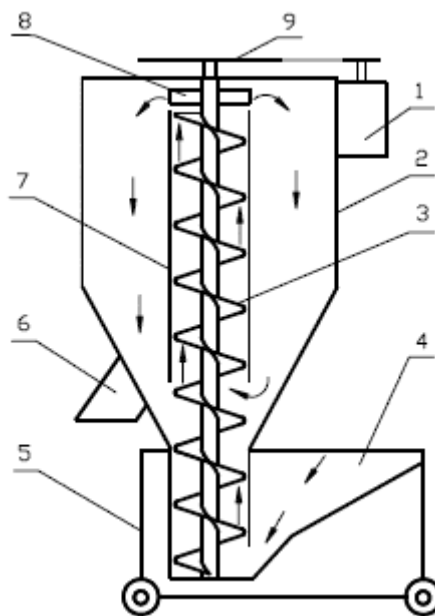
Tất cả các công đoạn đều được định lượng bằng cân thủ công theo công thức phối trộn, công nhân định lượng đồng thời là công nhân đứng máy trộn.

1.2.6.3. Công đoạn trộn

Nhiệm vụ: Trộn đều các thành phần đã định.

Công đoạn trộn được thực hiện bằng máy trộn hỗn hợp bột khô, trộn gián đoạn theo mẻ thực hiện trộn từng phần kiểu một trục vít thẳng đứng.

Cấu tạo máy trộn vít kiểu đứng:



Hình 1.5: Máy trộn vít đứng

1. Động cơ; 2. Vỏ thùng; 3. Vít trộn; 4. Cửa nạp liệu; 5. Khung máy;
6. Cửa tháo liệu; 7. Ống khuếch tán; 8. Cánh tung; 9. Puli truyền động cho vít.

Mô tả hoạt động:

Hỗn hợp được cung cấp vào máng cấp liệu (4) và được phần dưới của vít trộn (3) nâng lên ống khuấy tán (7) và đảo trộn. Khi hỗn hợp đi hết chiều cao của ống khuấy tán, nhờ lực ly tâm của cánh vít (3), hỗn hợp được đánh văng vào thùng trộn (2) và rơi trở lại xuống phần hình côn của thùng. Tại đây vật liệu lại được vít trộn nâng lên vào ống khuấy tán. Quá trình này được thực hiện lặp đi lặp lại nhiều lần và hỗn hợp được đảo trộn khá mạnh trong suốt thời gian trộn. Sau khi trộn, hỗn hợp được lấy ra qua cửa tháo liệu (6).

1.2.6.4. Công đoạn vận chuyển và bộ phận trung gian

Nhiệm vụ: Vận chuyển nguyên liệu, bán thành phẩm và thành phẩm theo quá trình công nghệ.

- a. Vít tải đứng: Nạp hỗn hợp đã trộn vào si lô chứa chờ ép viên
- b. Si lô chứa thức ăn hỗn hợp chờ ép viên
- c. Băng tải nghiêng

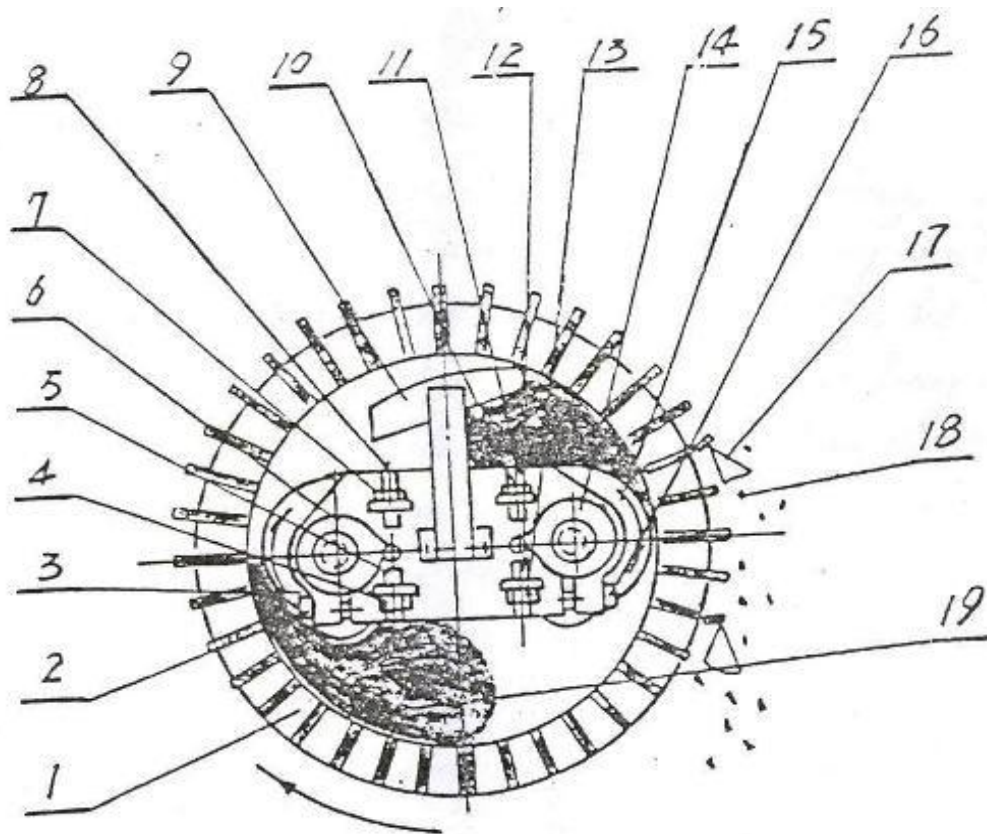
1.2.6.5. Công đoạn ép viên

Nhiệm vụ: tạo hình viên thức ăn theo kích thước và hình dáng qui định, thỏa mãn yêu cầu kỹ thuật chăn nuôi.

Cấu tạo buồng ép viên:

Máy ép tạo viên gồm có trục đặc trong, trục rỗng bao ở ngoài, nghĩa là 2 trục lồng vào nhau. Trục rỗng có 2 ổ bi, vòng ngoài của ổ bi lắp vào 2 thân ổ lắp chặt vào thành máy. Một đầu trục có mặt bích để lắp khuôn ép. Khi trục rỗng quay thì khuôn ép quay theo tốc độ quay của khuôn phải căn cứ vào đặc tính của nguyên liệu và căn cứ vào đường kính của viên để chọn cho phù hợp. theo kinh nghiệm thì với khuôn ép có đường kính lỗ bé thì phải sử dụng tốc độ tiếp tuyến tương đối cao, còn với khuôn có đường kính lỗ khuôn lớn thì phải sử dụng vận tốc tiếp tuyến tương đối thấp. Vận tốc tiếp tuyến của khuôn có ảnh hưởng đến hiệu suất tạo viên, đến tiêu hao năng lượng và độ chắc của viên.

Trong phạm vi nhất định, vận tốc của tiếp tuyến của khuôn cao thì năng suất cao, năng lượng tiêu hao cao, độ cứng của viên va chỉ số tỉ lệ hồ hoá bột cũng tăng lên. Nói chung, với đường kính lỗ khuôn là $3,2 \div 6,4\text{mm}$ thì vận tốc tiếp tuyến của khuôn rất cao có thể đạt tới $10,2\text{m/s}$; còn khi đường kính lỗ khuôn $16 \div 19\text{mm}$ thì vận tốc tiếp tuyến của khuôn ép là $6,1 \div 6,6\text{m/s}$. Nếu sử dụng 1 loại vận tốc tiếp tuyến để sản xuất đa dạng loại thức ăn là không tốt, chẳng hạn khi sử dụng máy ép viên cỡ lớn để sản xuất thức ăn viên có đường kính nhỏ thì chất lượng và hiệu quả không tốt như sử dụng máy ép viên cỡ nhỏ: đặc biệt rõ ràng nhất là khi sản xuất thức ăn chăn nuôi gia cầm và thức ăn chăn nuôi thủy sản có đường kính 3 mm. Nguyên nhân là vận tốc tiếp tuyến của khuôn ép quá ép còn đường kính của quả lô ép lại quá lớn tạo cho nguyên liệu ép qua lỗ quá nhanh, từ đó khiến cho chỉ số độ cứng và tỉ lệ hồ hoá của bột bị ảnh hưởng để khắc phục tình trạng nói trên nhằm thích ứng với nhu cầu ra công cho các nguyên liệu và các đường kính lỗ khuôn khác nhau ở nước ngoài nhiều công ty kết cấu $2 \div 3$ tốc độ tiếp tuyến khác nhau của khuôn trên một máy. Trục đặc không quay và được lắp ổ đỡ trên đó một đầu của trục đặc có 1 mặt bích. Trên mặt bích đó được lắp 2 hoặc 3 quả lô ép. Quả lô ép quay tròn quanh mình nó khe hở giữa quả lô ép với khuôn ép phải điều chỉnh thích hợp mới ép tạo thành viên được khế hở này nói chung là từ 0,1 đến 0,3. Nguyên lý làm việc của buồng ép viên (xem hình 1.6)



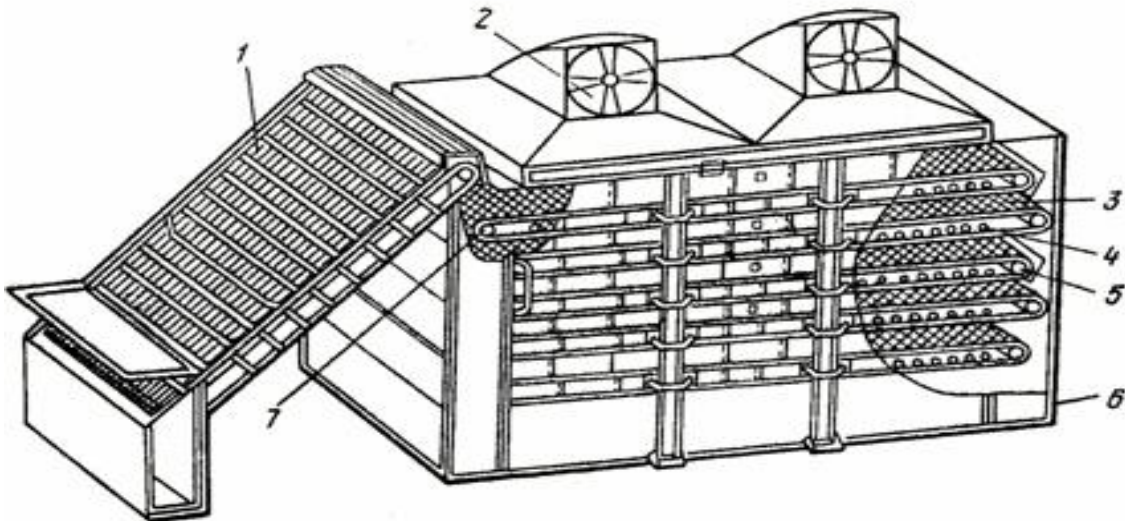
Hình 1.6 : Nguyên lý làm việc của buồng ép

- 1: Khuôn ép
- 2,16: Bulông kẹp chặt
- 3,15: Quả lô ép
- 4,7,10,13: Đai ốc chống nới lỏng
- 5,8,11,12: Bulông điều chỉnh
- 6,14: Bánh răng điều tiết
- 9: Dao gạt liệu vào
- 17: Dao cắt viên
- 18: Viên
- 19: Khu vực vật liệu để tạo viên

1.2.6.6. Công đoạn sấy và làm nguội

Nhiệm vụ: Làm khô thức ăn đạt đến độ ẩm cần thiết và làm nguội để đưa đến công đoạn cuối cùng là đóng gói sản phẩm.

Cấu tạo thiết bị sấy băng tải:



Hình 1.7: Máy sấy băng tải

Thiết bị máy sấy băng tải là một thiết bị sấy lí tưởng, nó được ứng dụng rộng rãi trong sấy khô các nguyên liệu thái lát, nguyên liệu có sọc, nguyên liệu hình dạng cục, và các loại nguyên liệu hạt ngũ cốc...chủ yếu dùng trong các ngành công nghiệp như chế biến lương thực thực phẩm, đông dược, nông nghiệp, thức ăn chăn nuôi...

Thiết bị dùng nguồn nhiệt trung gian là hơi nóng, nó dẫn hơi nóng trao đổi đầy đủ với sản phẩm ướt. Nó không những bằng sự đối lưu dẫn hơi nóng đi qua sản phẩm sấy mà còn lấy đi hơi nước bay hơi. Theo các chuyên gia nghiên cứu cho rằng, với kiểu máy sấy khô đối lưu, tốc độ sấy khô tăng 2-3 lần so với kiểu sấy khô hơi nóng nằm ngang...

Thiết bị này có đặc tính là hiệu suất cao, tiết kiệm nhiên liệu. Nó chính là loại máy sấy có chất lượng hàng đầu vì nó được thiết kế theo kiểu hơi nóng thấm sâu, cấp khí nóng theo từng tầng, nhiều mức nhiệt độ, tuần hoàn nhiệt...

Băng tải nhiều tầng kiểu lưới bằng thép không gỉ, sản phẩm sấy được dàn trải trên đó, Khí nóng xuyên qua băng tải lưới và sản phẩm sấy theo chiều từ dưới lên hoặc từ trên xuống, khí nóng trao đổi nhiệt rất đồng đều và hiệu suất cao, năng suất sấy có tỉ lệ cao và chất lượng sản phẩm tốt.

Thiết bị được lắp hệ thống băng tải tiếp liệu và có bộ dàn trải nguyên liệu đảm bảo cho độ dày của nguyên liệu sấy được trải đều lên băng tải sấy.

Mô tả hoạt động:

Hình 1.7 miêu tả máy sấy băng tải là tủ kim loại kín, bên trong có từ 4-5 nhánh băng tải 3. Các băng chuyền và mỗi băng được sản xuất bằng lưới thép không gỉ được căng ra trên tang truyền chủ động 7 và tang bị động 5. Các băng tải có bề rộng khác nhau phụ thuộc vào năng suất mỗi máy sấy. Mỗi băng có thể có bộ dẫn động lập với hộp giảm tốc, hoặc có thể có bộ dẫn động chung cho phép thay đổi tốc độ của các băng tải từ 1.14 đến 1.0m/phút. Không khí để sấy cho vào dưới nhánh thứ 2 của băng tải và được đun nóng nhờ các calorife hơi 4 lắp giữa các băng lưới của mỗi nhánh. Không khí xuyên qua tất cả các băng lưới và sản phẩm nằm trên đó. Không khí được bão hòa ẩm và sau khi làm vô trùng thì được quạt 2 thổi ra ngoài.

Sản phẩm trước khi sấy cần tán nhỏ sơ bộ và băng tải 1 chuyển đến nhánh trên của băng chuyền máy sấy. Sản phẩm cùng với băng chuyền đến đầu cuối cùng rồi đổ xuống băng dưới.

Chương 2.

CÁC PHẦN TỬ SỬ DỤNG TRONG HỆ THỐNG SẤY BĂNG TẢI

2.1. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN ATMEGA8

2.1.1. Tổng quan về vi điều khiển ATMEGA8



Hình 2.1: Hình dạng Atmega8

Các tính năng:

Hiệu suất cao, tiết kiệm điện

Nâng cao cấu trúc RISC

- 130 lệnh hiệu quả
- 32 x 8 thanh ghi chung đa năng + các thanh ghi điều khiển ngoại vi
- Đầy đủ các quá trình điều khiển tĩnh
- Lên đến 16MIPS dữ liệu tại 16MHz
- Chip 2 nhân

Độ bền, sức chịu đựng cao, không thay đổi phân vùng nhớ

- 8KBytes bộ nhớ Flash có thể lập trình trong hệ thống
- 512Bytes EEPROM
- 1KByte bộ nhớ SRAM bên trong
- Chu kỳ Ghi / Xoá bỏ: 10.000 Flash/100, 000 EEPROM
- Độ bền dữ liệu: 20 năm ở 85 độ C/100 năm ở 25 độ C (1)
- Đoạn mã lựa chọn chế độ khởi động với các bit khóa độc lập trong chương trình hệ thống bởi chương trình khởi động đọc thật trong khi quá trình ghi diễn ra
- Khóa Lập trình Phần mềm bảo mật

Thiết bị ngoại vi

- 2 bộ Timer/counter 8 bit với bộ đếm gộp trước riêng biệt và chế độ so sánh mẫu
- 2 bộ Timer/counter 16 bit mở rộng với bộ đếm gộp trước chế độ so sánh mẫu và chế độ thu thập (bắt dữ liệu)
- Bộ counter thời gian thực với bộ dao động (oscillator) riêng biệt
- 3 kênh PWM
- 8-kênh ADC trong gói TQFP và QFN / MLF

Tám kênh 10-bit Độ chính xác

- 6-kênh ADC trong gói PDIP

Sáu kênh 10-bit Độ chính xác

- Byte-định hướng với 2 dây giao diện nối tiếp
- Lập trình kép các USART nối tiếp
- Giao diện nối tiếp SPI chủ tớ
- Lập trình Watchdog Timer bộ dao động trên chip riêng biệt
- Bộ so sánh tương tự trên chip

Các tính năng đặc biệt của bộ vi xử lý

- Thiết lập bật lại nguồn và lập trình lại khi phát hiện nguồn yếu (brown-out)
- Hiệu chỉnh bộ dao động RC bên trong
- Ngắt nguồn trong và ngoài
- 5 chế độ chờ (sleep): Idle, giảm ồn ADC, tiết kiệm điện (power-saver), ngắt điện (Power-down), và chế độ chờ (standby)

Cổng vào ra và dạng đóng gói

- 23 đường vào ra lập trình được
- 28 chân -PDIP, 32 chân-TQFP, và 32 khối QFN / MLF

Điện áp hoạt động

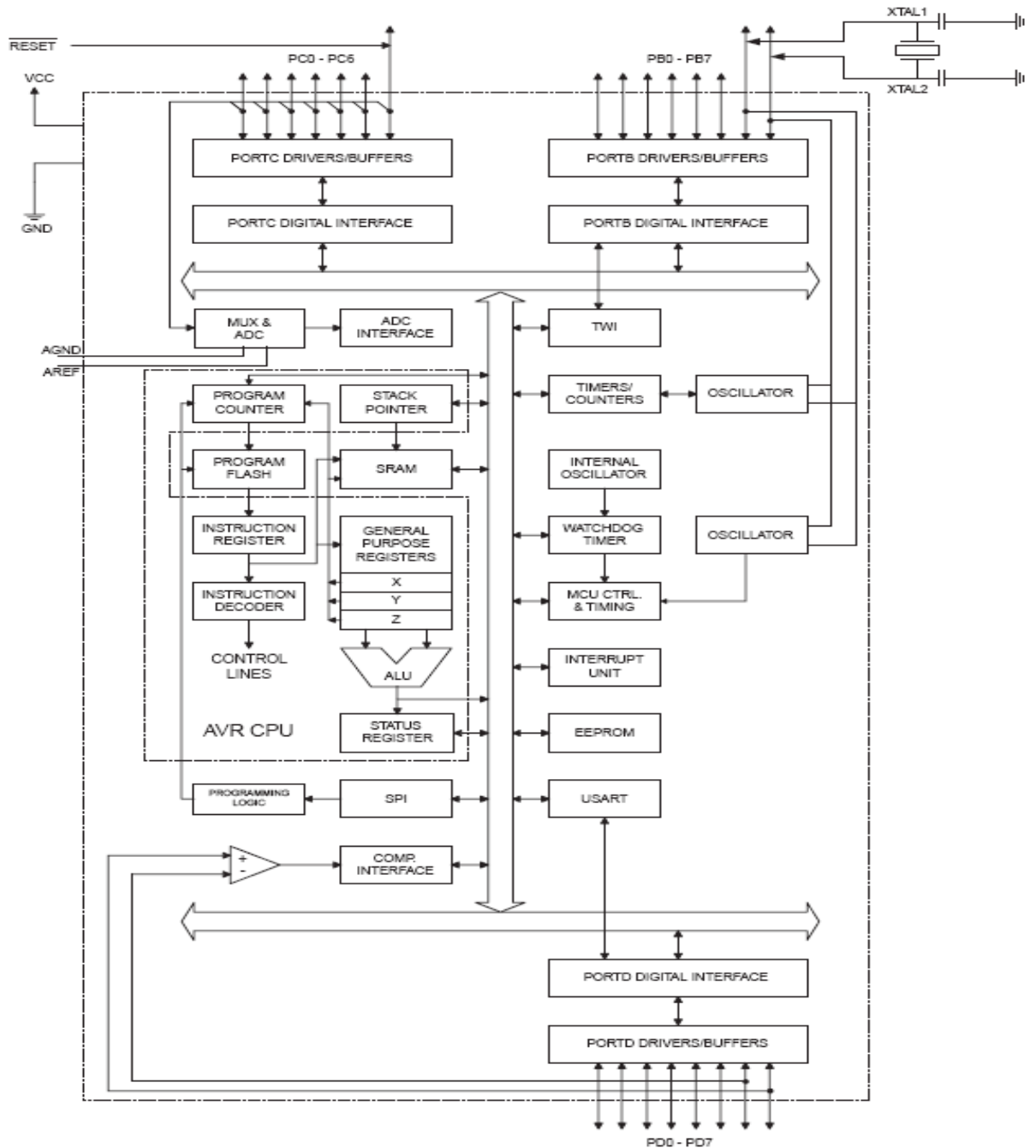
- 2,7 - 5.5V
- 0 - 16MHz

Công suất tiêu thụ ở 4MHz, 3V, 25 độ C

- Chế độ hoạt động: 3.6mA
- Chế độ chờ: 1.0mA
- Chế độ ngắt: 0.5 μ A

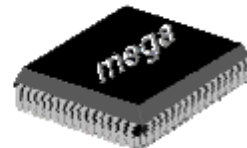
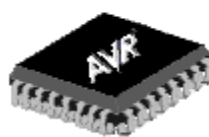
Vi điều khiển AVR do hãng Atmel (Hoa Kỳ) sản xuất được giới thiệu lần đầu năm 1996. AVR có rất nhiều dòng khác nhau bao gồm dòng Tiny AVR (như AT tiny 13, AT tiny 22...) có kích thước bộ nhớ nhỏ, ít bộ phận ngoại vi, rồi đến dòng AVR (chẳng hạn AT90S8535, AT90S8515,...) có kích thước bộ nhớ vào loại trung bình và mạnh hơn là dòng Mega (như ATmega32, ATmega128,...) với bộ nhớ có kích thước vài Kbyte đến vài trăm Kb cùng với các bộ ngoại vi đa dạng được tích hợp trên chip, cũng có dòng tích hợp cả bộ LCD trên chip (dòng LCD AVR). Tốc độ của dòng Mega cũng cao hơn so với các dòng khác.

Sự khác nhau cơ bản giữa các dòng chính là cấu trúc ngoại vi, còn nhân thì vẫn như nhau.



Hình 2.2: Sơ đồ khối ATmega8

Đặc biệt, năm 2008, Atmel lại tiếp tục cho ra đời dòng AVR mới là XmegaAVR, với những tính năng mạnh mẽ chưa từng có ở các dòng AVR trước đó. Có thể nói XmegaAVR là dòng MCU 8 bit mạnh mẽ nhất hiện nay.



Hình 2.3: Các loại AVR

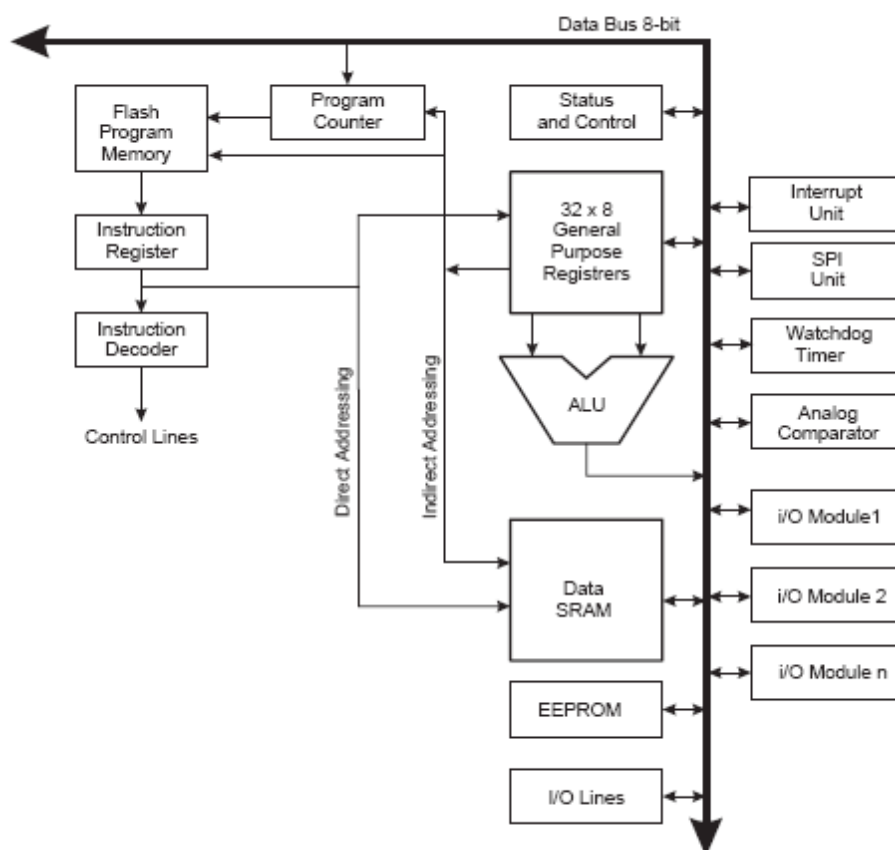
2.1.1.1. Các dòng AVR khác nhau: Tiny, AVR và Mega

Bộ Nhớ Dữ Liệu : Bộ nhớ dữ liệu của AVR chia làm 2 phần chính là bộ nhớ SRAM và bộ nhớ EEPROM. Tuy cùng là bộ nhớ dữ liệu nhưng hai bộ nhớ này lại tách biệt nhau và được đánh địa chỉ riêng.

Bộ nhớ SRAM có dung lượng 1 K bytes, Bộ nhớ SRAM có hai chế độ hoạt động là chế độ thông thường và chế độ tương thích với ATmega103, muốn thiết lập bộ nhớ SRAM hoạt động theo chế độ nào ta sử dụng bit cầu chì M103C (M103C fuse bit (9)).

Atmega8 là vi điều khiển 8 bit dựa trên kiến trúc RISC. Với khả năng thực hiện mỗi lệnh trong vòng một chu kỳ xung clock, Atmega8 có thể đạt được tốc độ 1MIPS trên mỗi MHz (1 triệu lệnh/s/MHz).

Dưới đây là sơ đồ khối của Atmega8



Hình 2.4: Sơ đồ cấu trúc Atmega8

ATmega 8 hỗ trợ đầy đủ các chương trình và công cụ phát triển hệ thống như: trình dịch C, macro assemblers, chương trình mô phỏng/sửa lỗi, kit thử nghiệm,...

2.1.2. Cấu trúc chung AVR

CPU của AVR có chức năng bảo đảm sự hoạt động chính xác của các chương trình. Do đó nó phải có khả năng truy cập bộ nhớ, thực hiện các quá trình tính toán, điều khiển các thiết bị ngoại vi và quản lý ngắt.

2.1.2.1. Cấu trúc tổng quát

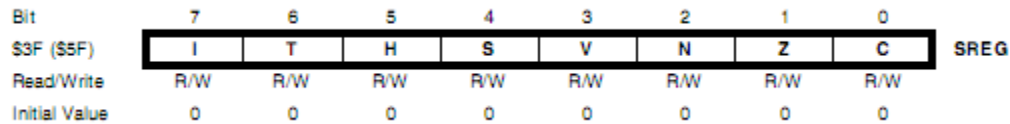
AVR sử dụng cấu trúc Harvard, tách riêng bộ nhớ và các bus cho chương trình và dữ liệu. Các lệnh được thực hiện chỉ trong một chu kỳ xung clock. Bộ nhớ chương trình được lưu trong bộ nhớ Flash.

2.1.2.2. ALU

ALU làm việc trực tiếp với các thanh ghi chức năng chung. Các phép toán được thực hiện trong một chu kỳ xung clock. Hoạt động của ALU được chia làm 3 loại: đại số, logic và theo bit.

2.1.2.3. Thanh ghi trạng thái

Đây là thanh ghi trạng thái có 8 bit lưu trữ trạng thái của ALU sau các phép tính số học và logic.



Hình 2.5: Thanh ghi trạng thái SREG

C: Carry Flag ;cờ nhớ (Nếu phép toán có nhớ cờ sẽ được thiết lập)

Z: Zero Flag ;Cờ zero (Nếu kết quả phép toán bằng 0)

N: Negative Flag (Nếu kết quả của phép toán là âm)

V: Two's complement overflow indicator (Cờ này được thiết lập khi tràn số bù 2)

V, For signed tests (S=N XOR V) S: N

H: Half Carry Flag (Được sử dụng trong một số toán hạng sẽ được chỉ rõ sau)

T: Transfer bit used by BLD and BST instructions(Được sử dụng làm nơi chung gian trong các lệnh BLD,BST).

I: Global Interrupt Enable/Disable Flag (Đây là bit cho phép toàn cục ngắt. Nếu bit này ở trạng thái logic 0 thì không có một ngắt nào được phục vụ.)

2.1.2.4. Các thanh ghi chức năng chung

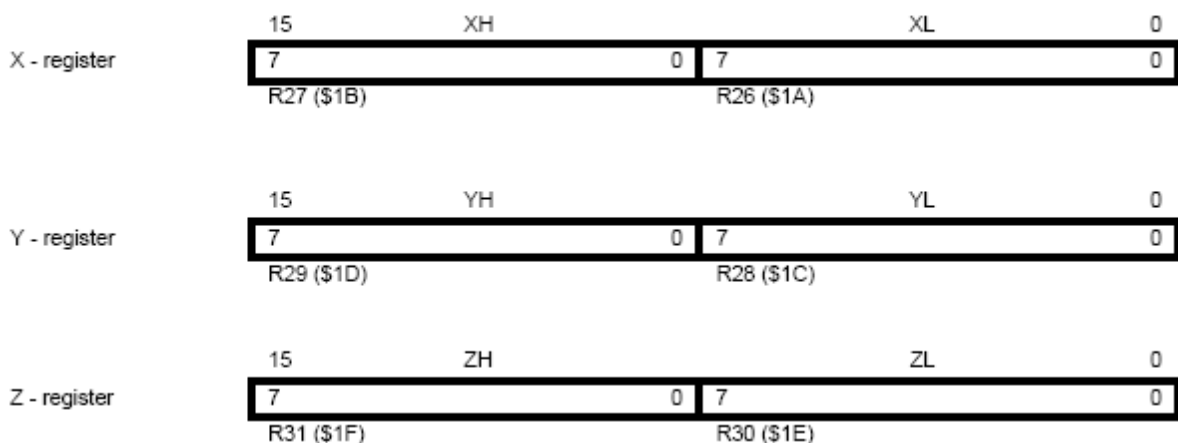
	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register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

Hình 2.6: Thanh ghi chức năng chung

Tập thanh ghi (register file) : Tập 32 thanh ghi đa chức năng (\$0000 - \$001F)

Đã được nói ở trên, ngoài chức năng là các thanh ghi đa chức năng, thì các thanh ghi từ R26 tới R31 từng đôi một tạo thành các thanh ghi 16 bit X, Y, Z được dùng làm con trỏ tới bộ nhớ chương trình và bộ nhớ dữ liệu. Thanh ghi con trỏ X, Y có thể dùng làm con trỏ tới bộ nhớ dữ liệu, còn thanh ghi Z có thể dùng làm con trỏ tới bộ nhớ chương trình. Các trình biên dịch C thường dùng các thanh ghi con trỏ này để quản lý **Data stack** của chương trình C.

Figure 5. The X-, Y-, and Z-registers



Hình 2.7: Chức năng con trỏ của các thanh ghi R26 –R31

2.1.2.5. Con trỏ ngăn xếp (SP)

Là một thanh ghi 16 bit nhưng cũng có thể được xem như hai thanh ghi chức năng đặc biệt 8 bit. Có địa chỉ trong các thanh ghi chức năng đặc biệt là \$3E (Trong bộ nhớ RAM là \$5E). Có nhiệm vụ trỏ tới vùng nhớ trong RAM chứa ngăn xếp.

Bit	15	14	13	12	11	10	9	8	
\$3E (\$5E)	-	-	-	-	-	-	SP9	SP8	SPH
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R	R	R	R	R	R	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	0	0	0	0	
	0	0	0	0	0	0	0	0	

Hình 2.8: Thanh ghi con trỏ ngăn xếp

Khi chương trình phục vụ ngắt hoặc chương trình con thì con trỏ PC được lưu vào ngăn xếp trong khi con trỏ ngăn xếp giảm hai vị trí. Và con trỏ ngăn xếp sẽ giảm 1 khi thực hiện lệnh push. Ngược lại khi thực hiện lệnh POP thì con trỏ ngăn xếp sẽ tăng 1 và khi thực hiện lệnh RET hoặc RETI thì con trỏ ngăn xếp sẽ tăng 2. Như vậy con trỏ ngăn xếp cần được chương trình đặt trước giá trị khởi tạo ngăn xếp trước khi một chương trình con được gọi hoặc các ngắt được cho phép phục vụ. Và giá trị ngăn xếp ít nhất cũng phải lớn hơn hoặc bằng 60H (0x60) vì 5FH trở lại là vùng các thanh ghi.

2.1.3. Cấu trúc ngắt của ATMEGA8

2.1.3.1. Khái niệm về ngắt

Ngắt là một cơ chế cho phép thiết bị ngoại vi báo cho CPU biết về tình trạng sẵn sàng cho đổi dữ liệu của mình. Ví dụ: Khi bộ truyền nhận UART nhận được một byte nó sẽ báo cho CPU biết thông qua cờ RXC, hoặc khi nó đã truyền được một byte thì cờ TX được thiết lập...

Khi có tín hiệu báo ngắt CPU sẽ tạm dừng công việc đang thực hiện lại và lưu vị trí đang thực hiện chương trình (con trỏ PC) vào ngăn xếp sau đó trỏ tới vector phục vụ ngắt và thực hiện chương trình phục vụ ngắt đó chờ tới khi gặp lệnh RETI (return from interrupt) thì CPU lại lấy PC từ ngăn xếp ra và tiếp tục thực hiện chương trình mà trước khi có ngắt nó đang thực hiện. Trong trường hợp mà có nhiều ngắt yêu cầu cùng một lúc thì CPU sẽ lưu các cờ báo ngắt đó lại và thực hiện lần lượt các ngắt theo mức ưu tiên. Trong khi đang thực hiện ngắt mà xuất hiện ngắt mới thì sẽ xảy ra hai trường hợp. Trường hợp ngắt

này có mức ưu tiên cao hơn thì nó sẽ được phục vụ. Còn nó mà có mức ưu tiên thấp hơn thì nó sẽ bị bỏ qua.

Bộ nhớ ngăn xếp là vùng bất kì trong SRAM từ địa chỉ 0x60 trở lên. Để truy nhập vào SRAM thông thường thì ta dùng con trỏ X,Y,Z và để truy nhập vào SRAM theo kiểu ngăn xếp thì ta dùng con trỏ SP. Con trỏ này là một thanh ghi 16 bit và được truy nhập như hai thanh ghi 8 bit chung có địa chỉ :SPL:0x3D/0x5D(IO/SRAM) và SPH:0x3E/0x5E.

Khi chương trình phục vụ ngắt hoặc chương trình con thì con trỏ PC được lưu vào ngăn xếp trong khi con trỏ ngăn xếp giảm hai vị trí. Và con trỏ ngăn xếp sẽ giảm 1 khi thực hiện lệnh push. Ngược lại khi thực hiện lệnh POP thì con trỏ ngăn xếp sẽ tăng 1 và khi thực hiện lệnh RET hoặc RETI thì con trỏ ngăn xếp sẽ tăng 2. Như vậy con trỏ ngăn xếp cần được chương trình đặt trước giá trị khởi tạo ngăn xếp trước khi một chương trình con được gọi hoặc các ngắt được cho phép phục vụ. Và giá trị ngăn xếp ít nhất cũng phải lớn hơn 60H (0x60) vì 5FH trở lại là vùng các thanh ghi.

2.1.3.2. Trình phục vụ ngắt và bảng vector ngắt

Đối với mỗi ngắt thì phải có một trình phục vụ ngắt ISR (Interrupt Service Routine) hay trình quản lý ngắt (Interrupt handler). Khi một ngắt được gọi thì bộ vi điều khiển phục vụ ngắt. Khi một ngắt được gọi thì bộ vi điều khiển chạy trình phục vụ ngắt. Đối với mỗi ngắt thì có một vị trí cố định trong bộ nhớ để giữ địa chỉ ISR của nó. Nhóm các vị trí nhớ được dành riêng để giữ các địa chỉ của các ISR được gọi là **bảng véc tơ ngắt**.

Khi kích hoạt một ngắt bộ vi điều khiển đi qua các bước sau:

- Vi điều khiển kết thúc lệnh đang thực hiện và lưu địa chỉ của lệnh kế tiếp (PC)

vào ngăn xếp.

- Nó nhảy đến một vị trí cố định trong bộ nhớ được gọi là bảng véc tơ ngắt nơi lưu

giữ địa chỉ của một trình phục vụ ngắt.

- Bộ vi điều khiển nhận địa chỉ ISR từ bảng véc tơ ngắt và nhảy tới đó. Nó bắt đầu

thực hiện trình phục vụ ngắt cho đến lệnh cuối cùng của ISR là RETI (trở về từ ngắt).

- Khi thực hiện lệnh RETI bộ vi điều khiển quay trở về nơi nó đã bị ngắt. Trước hết

nó nhận địa chỉ của bộ đếm chương trình PC từ ngăn xếp bằng cách kéo hai byte trên đỉnh của ngăn xếp vào PC. Sau đó bắt đầu thực hiện các lệnh từ địa chỉ đó.

2.1.3.3. Bảng vector ngắt

Table 18. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	SPM_RDY	Store Program Memory Ready

Hình 2.9: Bảng vector ngắt

2.1.3.4. Thứ tự ưu tiên ngắt

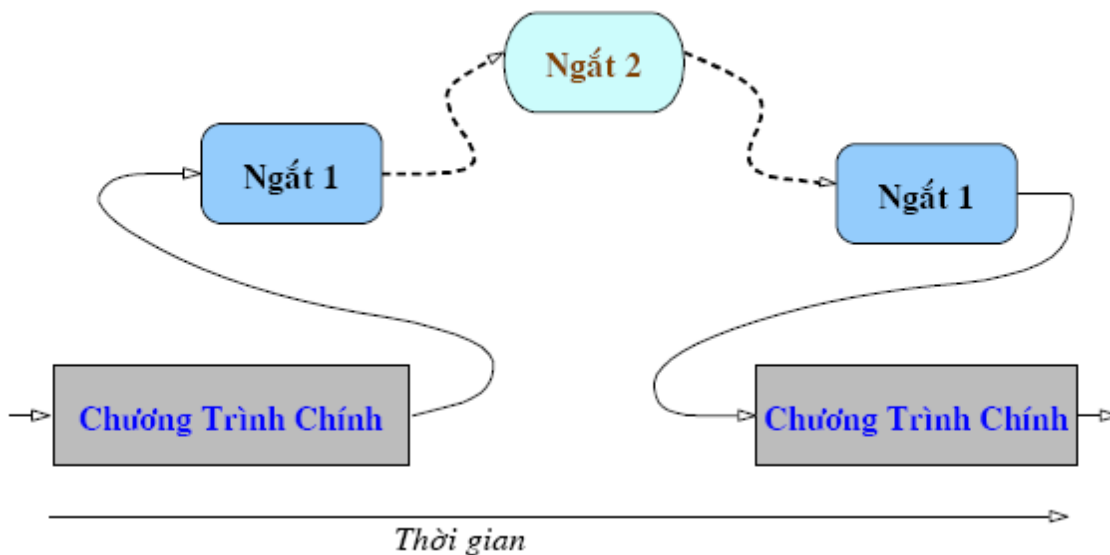
Không như vi điều khiển họ 8051, ở đó thứ tự ưu tiên của các ngắt có thể thay đổi được (bằng cách lập trình). Với vi điều khiển AVR thứ tự ưu tiên các ngắt là không thể thay đổi và theo qui tắc: “ **Một vec tơ ngắt có địa chỉ thấp hơn trong bộ nhớ chương trình có mức độ ưu tiên cao hơn** ”. Chặn hạn ngắt ngoài 0 (INT0) có mức độ ưu tiên cao hơn ngắt ngoài 1 (INT1).

Để cho phép một ngắt người dùng cần cho phép ngắt toàn cục (set bit I trong thanh ghi SREG) và các bit điều khiển ngắt tương ứng. Khi một ngắt xảy ra và đang được phục vụ thì bit I trong thanh ghi SREG bị xóa,

như thế khi có một ngắt khác xảy ra nó sẽ không được phục vụ, do đó để cho phép các ngắt trong khi một ISR (interrupt service routine) khác đang thực thi, thì trong chương trình ISR phải có lệnh SEI để set lại bit I trong SREG.

2.1.3.5. Ngắt trong ngắt

Khi AVR đang thực hiện một trình phục vụ ngắt thuộc một ngắt nào đó thì lại có một ngắt khác được kích hoạt. Trong những trường hợp như vậy thì một ngắt có mức ưu tiên cao hơn có thể ngắt một ngắt có mức ưu tiên thấp hơn. Lúc này ISR của ngắt có mức ưu tiên cao hơn sẽ được thực thi(*) . Khi thực hiện xong ISR của ngắt có mức ưu tiên cao hơn thì nó mới quay lại phục vụ tiếp ISR của ngắt có mức ưu tiên thấp hơn trước khi trở về chương trình chính. Đây gọi là ngắt trong ngắt



Hình 2.10 : Cấu trúc ngắt

Chú ý:

- Giả định là khi một ISR nào đó đang thực thi thì xảy ra một yêu cầu ngắt từ một
- ISR khác có mức ưu tiên thấp hơn thì ISR có mức ưu tiên thấp hơn không được phục vụ,
- nhưng nó sẽ không bị bỏ qua luôn mà ở trạng thái chờ. Nghĩa là ngay sau khi ISR có

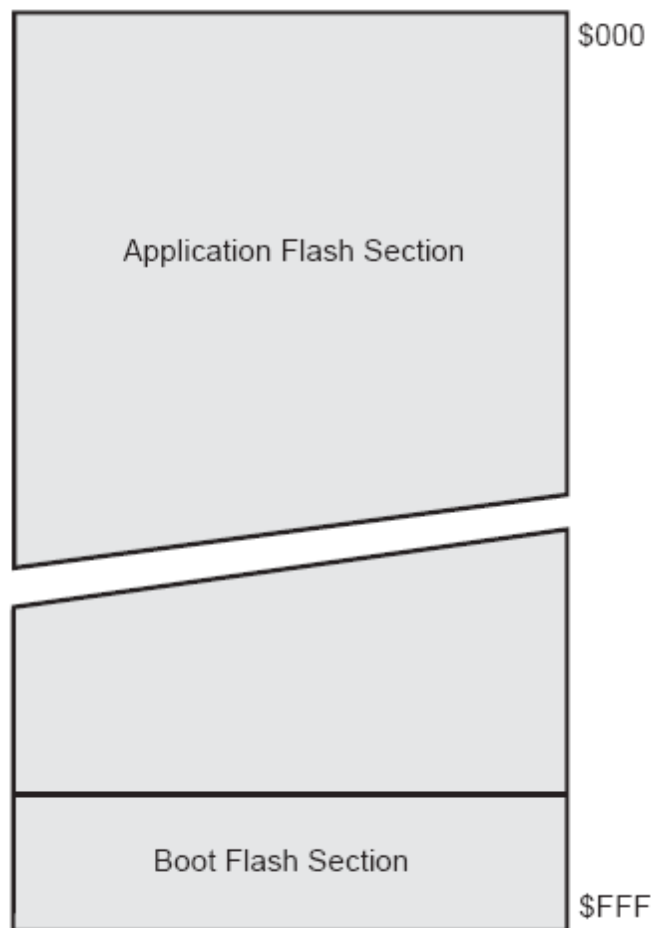
- mức ưu tiên cao hơn thực thi xong thì đến lượt ISR có mức ưu tiên thấp hơn sẽ được
- phục vụ.
- (*) : Điều này chỉ xảy ra khi trong code của ISR của ngắt có mức ưu tiên thấp
- hơn có lệnh set bit I trong thanh ghi SREG (đó là lệnh SEI).

2.1.4. Cấu trúc bộ nhớ

AVR có 2 không gian bộ nhớ chính là bộ nhớ dữ liệu và bộ nhớ chương trình. Ngoài ra Atmega8 còn có thêm bộ nhớ EEPROM để lưu trữ dữ liệu.

2.1.4.1. Bộ nhớ chương trình (Bộ nhớ Flash)

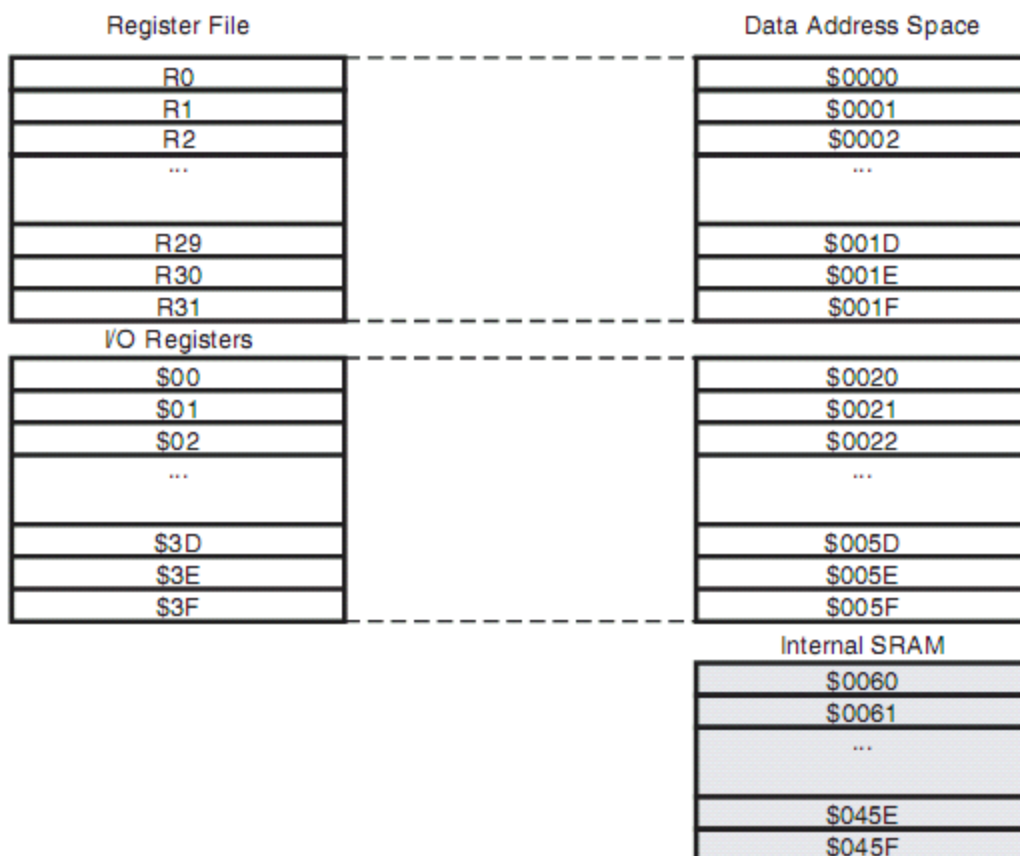
Bộ nhớ Flash 16KB của Atmega8 dùng để lưu trữ chương trình. Do các lệnh của AVR có độ dài 16 hoặc 32 bit nên bộ nhớ Flash được sắp xếp theo kiểu 8Kx16. Bộ nhớ Flash được chia làm 2 phần, phần dành cho chương trình boot và phần dành cho chương trình ứng dụng.



Hình 2.11: Bản đồ bộ nhớ chương trình

2.1.4.2. Bộ nhớ dữ liệu SRAM

1120 ô nhớ của bộ nhớ dữ liệu định địa chỉ cho file thanh ghi, bộ nhớ I/O và bộ nhớ dữ liệu SRAM nội. Trong đó 96 ô nhớ đầu tiên định địa chỉ cho file thanh ghi và bộ nhớ I/O, và 1024 ô nhớ tiếp theo định địa chỉ cho bộ nhớ SRAM nội.



Hình 2.2: Bản đồ bộ nhớ dữ liệu SRAM

2.1.4.3. Bộ nhớ dữ liệu EEPROM

Atmega8 chứa bộ nhớ dữ liệu EEPROM dung lượng 512 byte, và được sắp xếp theo từng byte, cho phép các thao tác đọc/ghi từng byte một. Đây là bộ nhớ dữ liệu có thể ghi xóa ngay trong lúc vi điều khiển đang hoạt động và không bị mất dữ liệu khi nguồn điện cung cấp bị cắt. Có thể ví bộ nhớ dữ liệu EEPROM giống như là ổ cứng (Hard disk) của máy vi tính. EEPROM được xem như là một bộ nhớ vào ra được đánh địa chỉ độc lập với SRAM, điều này có nghĩa là ta cần sử dụng các lệnh in, out ... khi muốn truy xuất tới EEPROM. Để điều khiển vào ra dữ liệu với EEPROM ta sử dụng 3 thanh ghi sau :

2.1.5. CÁC CÔNG VÀO RA (I/O)

Vi điều khiển ATmega8 có 23 đường vào ra chia làm 2 nhóm 8 bit, một nhóm 7 bit. Các đường vào ra này có rất nhiều tính năng và có thể lập trình được. Ở đây ta sẽ xét chúng là các công vào ra số. Nếu xét trên mặt này thì các công vào ra này là công vào ra hai chiều có thể định hướng theo từng bit. Và chứa cả điện trở pull-up (có thể lập trình được). Mặc dù mỗi port có các đặc điểm riêng nhưng khi xét chúng là các công vào ra số thì dường như điều khiển vào ra dữ liệu thì hoàn toàn như nhau. Chúng ta có thanh ghi và một địa chỉ công đối với mỗi công, đó là: thanh ghi dữ liệu công (PORTB, PORTC, PORTD), thanh ghi dữ liệu điều khiển công (DDRB, DDRC, DDRD) và cuối cùng là địa chỉ chân vào của công (PINB, PINC, PIND).

2.1.5.1. Các chức năng của Port B

a. XTAL2/TOSC2 – Port B, Bit 7

XTAL2: Chân 2 dao động tạo clock. Sử dụng chân clock thạch anh, hoặc dao động thạch anh tần số thấp. Khi dùng chân làm dao động thì không thể làm chân nhập xuất được nữa.

TOSC2: Chân 2 là dao động Timer. Nếu PB7 được dùng làm clock pin, DDB7, PORTB7 and PINB7 sẽ sẽ hiệu là mức 0

b. XTAL1/TOSC1 – Port B, Bit 6

XTAL1: Chip clock Oscillator pin 1.

TOSC1: Timer Oscillator pin 1.

Nếu PB6 dùng làm chân clock, DDB6, PORTB6 and PINB6 sẽ hiệu là mức 0.

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
PB5	SCK (SPI Bus Master clock Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2 (Timer/Counter2 Output Compare Match Output)
PB2	\overline{SS} (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output)
PB0	ICP1 (Timer/Counter1 Input Capture Pin)

Hình 2.13: Các chức năng Port B

c. SCK – Port B, Bit 5

SCK: Master Clock output, Slave Clock input pin for SPI channel. Khi SPI được kích hoạt là Slave, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB5.

d. MISO – Port B, Bit 4

MISO: Master Data input, Slave Data output pin for SPI channel. Khi SPI được kích hoạt là Master, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB4.

e. MOSI/OC2 – Port B, Bit 3

MOSI: SPI Master Data output, Slave Data input for SPI channel. Khi SPI được kích hoạt là Slave, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB3. Khi SPI được kích hoạt là Master, dữ liệu trực tiếp của chân này được điều khiển bởi DDB3.

f. \overline{SS} /OC1B – Port B, Bit 2

\overline{SS} : Slave Select ngõ vào. Khi SPI được kích hoạt là Slave, chân này được cấu hình là 1 chân ngõ vào bất chấp sự điều chỉnh từ DDB2.

g. OC1A – Port B, Bit 1

OC1A, Output Compare Match output: Chân PB1 có thể xử lý như 1 ngõ ra bên ngoài Timer/Counter1 Compare Match A.

h. ICP1 – Port B, Bit 0

ICP1 – chân giữ(chốt) ngõ vào : Chân PB0 có thể tác động làm 1 chân giữ cho Timer/Counter1.

2.1.5.2. Các chức năng của Port C

a. RESET – Port C, Bit 6

RESET, Reset pin: Khi cầu chì RSTDISBL đã lập trình, chức năng của chân này là vào ra bình thường, và 1 phần sẽ phải dựa vào Power-on Reset và Brown-out Reset như là nguồn reset của nó. Nếu chân PC6 dùng là chân reset, DDC6, PORTC6 và PINC6 sẽ hiểu là mức 0.

Port Pin	Alternate Function
PC6	RESET (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

Hình 2.14: Chức năng Port C

b. SCL/ADC5 – Port C, Bit 5

SCL, giao diện nối tiếp hai dây Xung nhịp: Khi bit TWEN trong TWCR set (one) để bật giao diện nối tiếp hai dây, pin PC5 bị ngắt từ port và trở thành chân Serial Clock I/O cho Two-wire Serial Interface.

c. SDA/ADC4 – Port C, Bit 4

SDA, Two-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the Two-wire Serial Interface, pin PC4 is disconnected from the port and becomes the Serial Data I/O pin for the Two-wire Serial Interface.

d. ADC3 – Port C, Bit 3

PC3 cũng có thể dùng là ADC input Channel 3. Chú ý là ADC input channel 3 dùng nguồn xoay chiều.

e. ADC2 – Port C, Bit 2

PC2 cũng có thể dùng là ADC input Channel 2. Chú ý là ADC input channel 2 dùng nguồn xoay chiều.

f. ADC1 – Port C, Bit 1

PC1 cũng có thể dùng là ADC input Channel 1. Chú ý là ADC input channel 1 dùng nguồn xoay chiều.

g. ADC0 – Port C, Bit 0

PC0 cũng có thể dùng là ADC input Channel 0. Chú ý là ADC input channel 0 dùng nguồn xoay chiều

2.1.5.3. Các chức năng của Port D

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input)
PD6	AIN0 (Analog Comparator Positive Input)
PD5	T1 (Timer/Counter 1 External Counter Input)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

Hình 2.15 : Chức năng Port C

a. AIN1 – Port D, Bit 7

AIN1, bộ so sánh tương tự thụ động ngõ vào. Cấu hình chân của port là nhập vào với ngắt pull-up bên trong để tránh nhiễu từ port số với chức năng của bộ so sánh tương tự.

b. AIN0 – Port D, Bit 6

AIN0, Bộ so sánh tương tự ngõ vào tích cực. Cấu hình chân của port là nhập vào với ngắt pull-up bên trong để tránh nhiễu từ port số với chức năng của bộ so sánh tương tự.

c. T1 – Port D, Bit 5

T1, số lượng mã nguồn Timer/Counter1.

d. XCK/T0 – Port D, Bit 4

XCK, USART xung nhịp ngoài. T0, số lượng mã nguồn Timer/Counter0.

e. INT1 – Port D, Bit 3

INT1, Ngắt nguồn bên ngoài 1: Chân PD3 có thể làm chức năng như 1 nguồn ngắt ngoài.

f. INT0 – Port D, Bit 2

INT0, Ngắt nguồn bên ngoài 0: Chân PD2 có thể làm chức năng như 1 nguồn ngắt ngoài.

g. TXD – Port D, Bit 1

TXD, Truyền tải dữ liệu (chân dữ liệu ra của USART). Khi bộ truyền USART được kích hoạt, chân này được cấu hình như là một ngõ ra bất kể giá trị của DDD1.

h. RXD – Port D, Bit 0

RXD, Nhận dữ liệu (chân dữ liệu vào của USART). Khi bộ nhận USART được kích hoạt, chân này được cấu hình như là một ngõ vào bất kể giá trị của DDD0

2.1.5.4. Mô tả thanh ghi của port I/O

The Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7 PORTB6 PORTB5 PORTB4 PORTB3 PORTB2 PORTB1 PORTB0								PORTB
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 2.16: Thanh ghi Port B

The Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDB7 DDB6 DDB5 DDB4 DDB3 DDB2 DDB1 DDB0								DDRB
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 2.17: Thanh ghi DDRB

The Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7 PINB6 PINB5 PINB4 PINB3 PINB2 PINB1 PINB0								PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Hình 2.18: Thanh ghi PINB

The Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
	- PORTC6 PORTC5 PORTC4 PORTC3 PORTC2 PORTC1 PORTC0								PORTC
Read/Write	R	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 2.19: Thanh ghi Port C

The Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	- DDC6 DDC5 DDC4 DDC3 DDC2 DDC1 DDC0								DDRC
Read/Write	R	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 2.20: Thanh ghi DDRC

The Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	PINC								
	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Hình 2.21: Thanh ghi PINC

The Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD								
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	
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 2.22: Thanh ghi Port D

The Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDRD								
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	
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 2.23: Thanh ghi DDRD

The Port D Input Pins Address – PIND

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

Hình 2.24: Thanh ghi PIND

Tóm lại:

1. Để đọc dữ liệu từ ngoài thì ta phải thực hiện các bước sau:

Đưa dữ liệu ra thanh ghi điều khiển DDR_{xn} để đặt cho PORT_x (hoặc bit n trong port) đó là đầu vào (xóa thanh ghi DDR_x hoặc bit).

Sau đó kích hoạt điện trở pull-up bằng cách set thanh ghi PORT_x (bit).

Cuối cùng đọc dữ liệu từ địa chỉ PIN_{xn} (trong đó x: là cổng và n là bit).

2. Để đưa dữ liệu từ vi điều khiển ra các cổng cũng có các bước hoàn toàn tương tự. Ban đầu ta cũng phải định nghĩa đó là cổng ra bằng cách set bit tương ứng của cổng đó....và sau đó là ghi dữ liệu ra bit tương ứng của thanh ghi PORT_x.

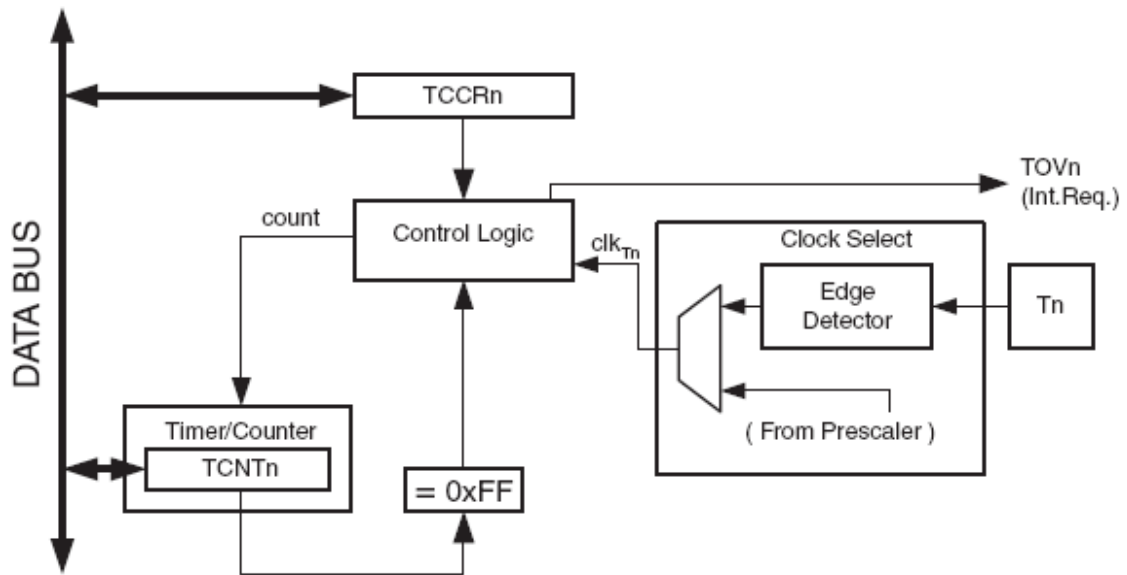
2.1.6. BỘ ĐỊNH THỜI 8BIT TIMER/COUNTER 0

Bộ định thời (timer/counter0) là một module định thời/đếm 8 bit, có các đặc điểm sau:

- Bộ đếm một kênh
- Xóa bộ định thời khi trong mode so sánh (tự động nạp)
- PWM
- Tạo tần số
- Bộ đếm sự kiện ngoài
- Bộ chia tần 10 bit
- Nguồn ngắt tràn bộ đếm và so sánh

AVR Atmega8 có tích hợp bộ timer/counter. Ta bắt đầu phần này bằng sơ đồ khối sau:

Sơ đồ cấu trúc của bộ định thời:



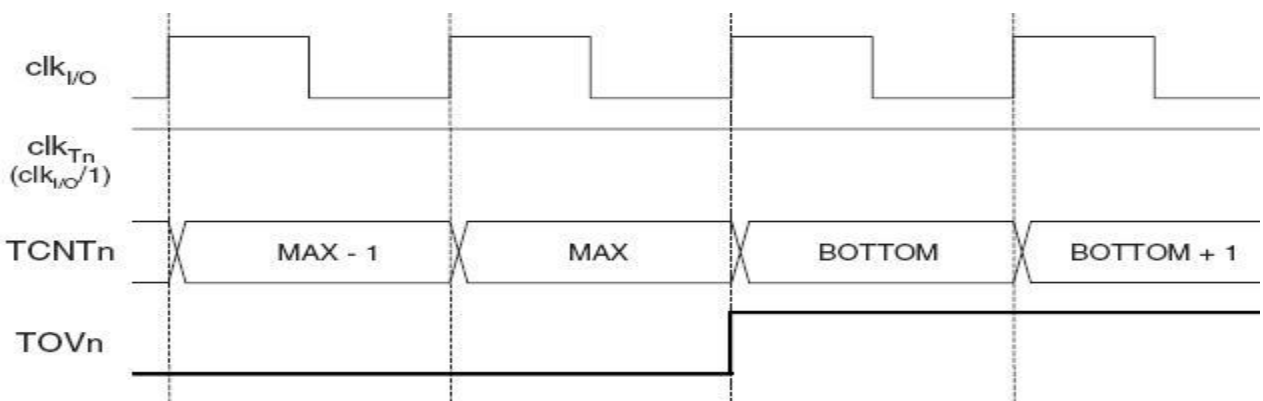
Hình 2.25: Sơ đồ cấu trúc bộ định thời

2.1.6.1. Hoạt động của bộ Timer/Counter

Mạch đếm lên làm thanh ghi TCNTn tăng 1 đơn vị mỗi khi có xung clk_{Tn} , khi đạt giá trị lớn nhất (8bit=255), cờ TOVn được set (logic 1) và bộ đếm tràn, giá trị bộ đếm TCNTn trở về 00 và tiếp tục đếm.

Xung clk_{Tn} có thể được lựa chọn từ nhiều nguồn khác nhau. Khi chọn xung nội (system clock), Timer/Counter là một Timer. Khi chọn xung ngoài (thông qua chân Tn) Timer/Counter là Counter.

Hoạt động này có thể diễn tả bằng giản đồ xung sau:



Hình 2.26: Giản đồ xung hoạt động

Cũng giống như bộ timer/counter trong các vi điều khiển khác, chúng ta quan tâm đến 2 thanh ghi: Timer/Counter Control và Timer/Counter Value. Trong AVR, đó là thanh ghi TCCRn và TCNTn.

Bit	7	6	5	4	3	2	1	0	
	TCCR0								
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$ /(No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Hình 2.27: Thanh ghi TCCRn

Clock Select Bit Description

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								
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 TCNTn

TCNT0 - Timer/C

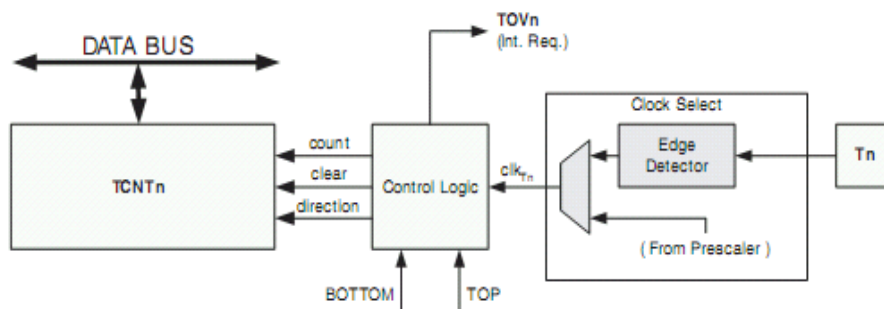
TCNT0 và OCR0 là các thanh ghi 8 bit. Các tín hiệu yêu cầu ngắt đều nằm trong thanh ghi TIFR. Các ngắt có thể được che bởi thanh ghi TIMSK.

Bộ định thời có thể sử dụng xung clock nội thông qua bộ chia hoặc xung clock ngoài trên chân T0. Khi chọn xung clock điều khiển việc bộ định thời/bộ đếm sẽ dùng nguồn xung nào để tăng giá trị của nó. Ngõ ra của khối chọn xung clock được xem là xung clock của bộ định thời (clk_{T0}).

Thanh ghi OCR0 luôn được so sánh với giá trị của bộ định thời/bộ đếm. Kết quả so sánh có thể được sử dụng để tạo ra PWM hoặc biến đổi tần số ngõ ra tại chân OC0.

2.1.6.2. Đơn vị đếm

Phần chính của bộ định thời 8 bit là một đơn vị đếm song hướng có thể lập trình được. Cấu trúc của nó như hình dưới đây:



Hình 2.29 : Đơn vị đếm

count: tăng hay giảm TCNT0 1

direction: lựa chọn giữa đếm lên và đếm xuống

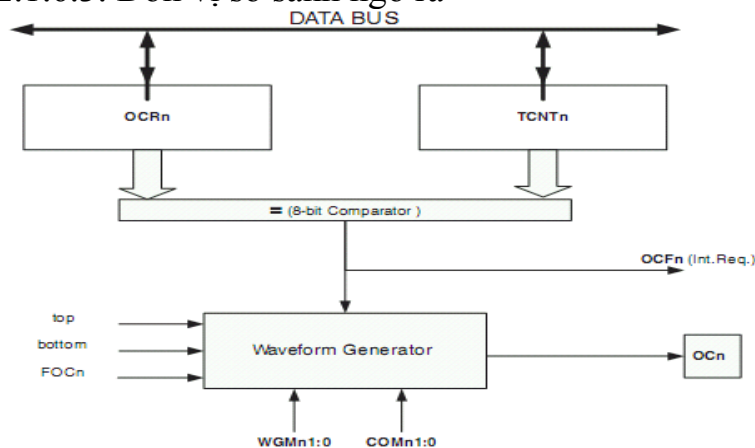
clear: xóa thanh ghi TCNT0

clk_{T0}: xung clock của bộ định thời

TOP: báo hiệu bộ định thời đã tăng đến giá trị lớn nhất

BOTTOM: báo hiệu bộ định thời đã giảm đến giá trị nhỏ nhất (0)

2.1.6.3. Đơn vị so sánh ngõ ra

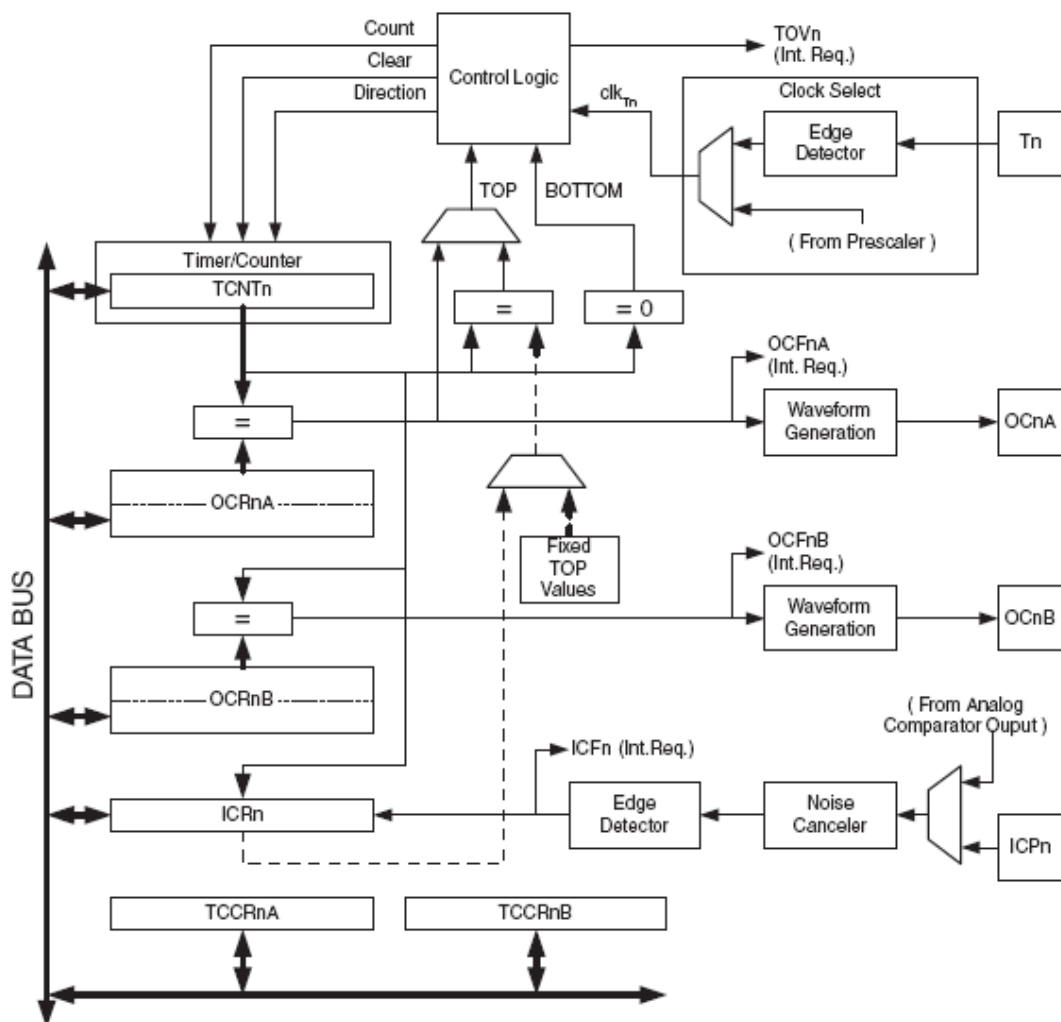


Hình 2.30: Sơ đồ đơn vị so sánh ngõ ra

Bộ so sánh 8 bit liên tục so sánh giá trị TCNT0 với giá trị trong thanh ghi so sánh ngõ ra (OCR0). Khi giá trị TCNT0 bằng với OCR0, bộ so sánh sẽ tạo một báo hiệu. Báo hiệu này sẽ đặt giá trị cờ so sánh ngõ ra (OCF0) lên 1 vào chu kỳ xung clock tiếp theo. Nếu được kích hoạt (OCIE0=1), cờ OCF0 sẽ tạo ra một ngắt so sánh ngõ ra và sẽ tự động được xóa khi ngắt được thực thi. Cờ OCF0 cũng có thể được xóa bằng phần mềm.

2.1.7. Bộ định thời/dếm TIMER/COUNTER 1 16-BIT

2.1.7.1. Sơ đồ khối và một số đặc điểm



Hình 2.31: Sơ đồ khối bộ định thời

Bộ định thời (timer/counter1) là một module định thời/dếm 16 bit, có các đặc điểm sau:

- True 16-bit Design (i.e., allows 16-bit PWM)

- 2 đơn vị ngõ vào so sánh độc lập(Two Independent Output Compare Units)
- đôi thanh ghi so sánh ngõ ra đệm(Double Buffered Output Compare Registers)
- 1 đơn vị chốt ngõ vào(One Input Capture Unit)
- Bộ chống nhiễu lối vào(Input Capture Noise Canceler)
- Xóa timer trong Compare Match (Clear Timer on Compare Match (Auto Reload))
- chống nhiễu sọc ngang(Glitch-free, Phase Correct Pulse Width Modulator (PWM))
- Giá trị chu kỳ PWM
- Bộ phát tần số chung
- Bộ đếm sự kiện ngoài
- 4 nguồn ngắt độc lập (TOV1, OCF1A, OCF1B, and ICF1)

2.1.7.2. Một số định nghĩa

BOTTOM Bộ đếm đạt tới *BOTTOM* khi có giá trị 0x0000

MAX Bộ đếm đạt tới *MAXimum* khi khi đạt giá trị 0xFFFF (decimal 65535).

TOP Bộ đếm đạt tới *TOP* khi nó bằng với giá trị lớn nhất của chuỗi đếm. Giá trị này có thể được gán bởi các giá trị cố định : 0x00FF, 0x01FF, or 0x03FF, hoặc giá trị trong bộ nhớ của các thanh ghi OCR1A ,ICR1 .

2.1.8. SPI(SERIAL PERIPHERAL INTERFACE)

Sơ đồ và định nghĩa

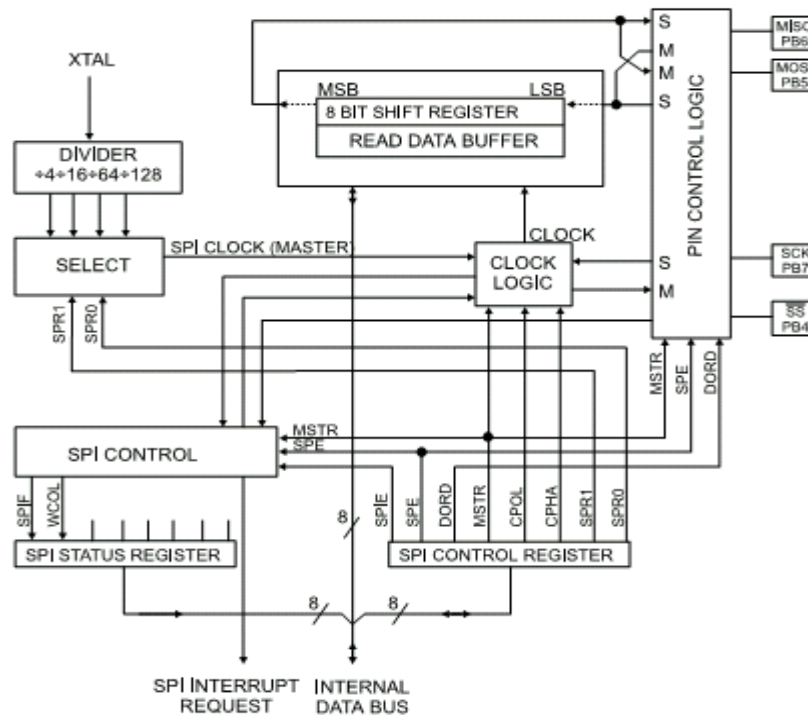
SPI là một giao diện thực hiện việc trao đổi dữ liệu giữa các thiết bị tương thích với khung dữ liệu 8bit và được truyền đồng bộ (cùng xung nhịp đồng hồ).

SPI cho phép truyền dữ liệu nối tiếp đồng bộ giữa thiết bị ngoại vi và vi điều khiển AVR hoặc giữa các vi điều khiển AVR. SPI của AT90S8535 có các đặc điểm đặc biệt sau:

- Chế độ song công, truyền dữ liệu đồng bộ 3 dây.
- Có thể giữ vai trò Master hoặc Slave.
- Bit MSB hoặc LSB có thể được truyền trước tùy vào người lập trình.

- Bốn tốc độ truyền có thể lập trình thông qua hai bit
- Cờ ngắt báo kết thúc truyền
- Vận hành từ trạng thái ngủ (Được đánh thức từ trạng thái ngủ).

Sơ đồ cấu trúc:



Hình 2.32: Sơ đồ cấu trúc SPI

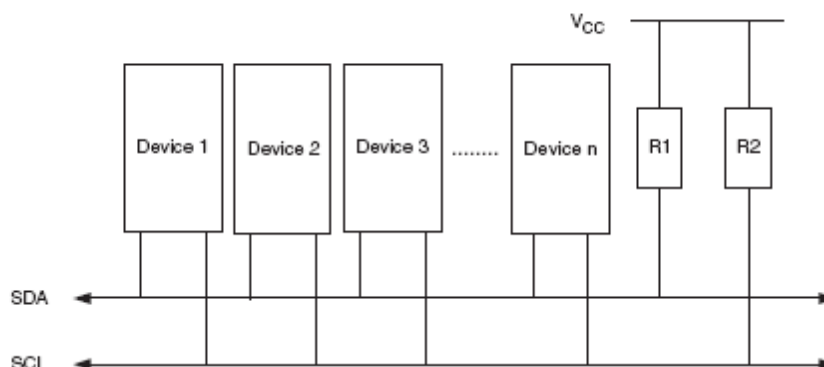
Để điều khiển khối giao tiếp SPI thì chúng ta có 3 thanh ghi. Đó là 1 thanh ghi điều khiển SPCR (SPI control Register), thanh ghi trạng thái SPSR (SPI status Register) và cuối cùng là thanh ghi dữ liệu SPDR (SPI Data Register).

2.1.9. TWI – TWO WIRE INTERFACE

Two Wire Interface là một sợi dây kết nối bus 2 chiều ,mà nó phù hợp với IC và SMBus.

Một thiết bị được kết nối đến một bus phải hành động như là một chủ hoặc thứ cấp.Đầu master thực hiện giai đoạn đầu cho sự vận chuyển dữ liệu với đầu salve trên bus,và hỏi xem nó có muốn vận chuyê hay nhận dữ liệu hay không.Một bus có thể có nhiều master,và một bộ xử lý điều phối ưu tiên ,nếu hai hoặc nhiều master cố gắng vận chuyển ở cùng một thời điểm.

Figure 68. TWI Bus Interconnection



Hình 2.33: Kết nối bus

Module TWi bao gồm bus chế độ logic mà nó có thể thu thập thông tin để tìm các điều kiện ngừng và bắt đầu ,bus bị ðụng ðộ và bus bị lỗi .Điều này có thể ðược sử dụng để xác ðịnh chế ðộ bus (chạy không ,chủ, ,bận hoặc không biết) trong kiểu master .Bus chế ðộ logic tiếp tục hoạt ðộng trong tất cả các chế ðộ nghỉ bao gồm chế ðộ nguồn giảm.

Thuật ngữTWI

The following definitions are frequently encountered in this section.

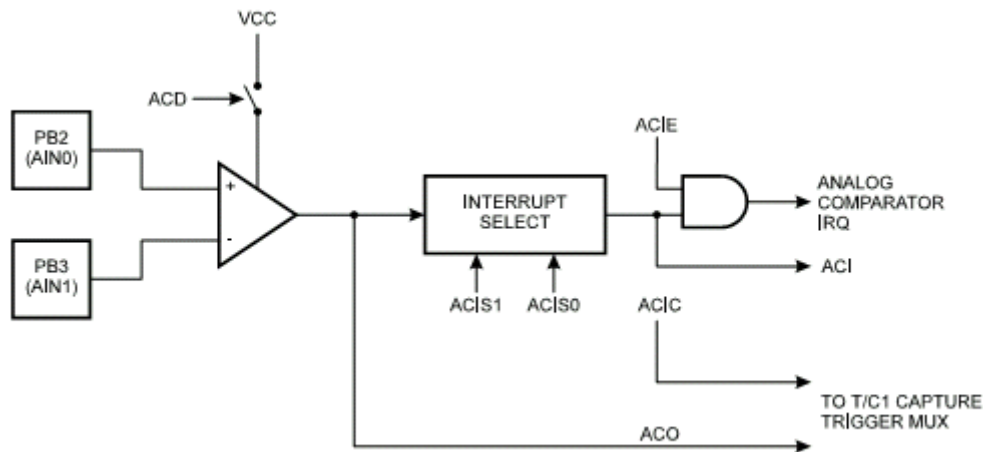
Table 64. TWI Terminology

Term	Description
Master	The device that initiates and terminates a transmission. The Master also generates the SCL clock.
Slave	The device addressed by a Master.
Transmitter	The device placing data on the bus.
Receiver	The device reading data from the bus.

Hình 2.34: Thuật ngữ TWI

2.1.10. Bộ so sánh tương tự (ALALOG COMPARATOR)

Bộ so sánh tương tự của AVR có đầu vào là hai chân PB2 và PB3 (như hình vẽ). Với chân PB2 ðược nối vào cực dương của bộ so sánh và PB3 ðược nối vào cực âm của bộ so sánh.Nó tạo ra hai mức logic nếu $V_{+} > V_{-}$ thì tín hiệu ra là 1 và ngược lại là 0.



Hình 2.35: Sơ đồ khối bộ so sánh tương tự

Để điều khiển và qua sát trạng thái của bộ so sánh tương tự ta có một thanh ghi đó là thanh ghi ACSR. Trước khi tìm hiểu về nguyên tắc hoạt động của nó ta sẽ giới thiệu về thanh ghi này.

Thanh ghi ACSR là một thanh ghi 8 bit có địa chỉ trong các thanh ghi I/O là 0x08 và có địa chỉ trong không gian bộ nhớ SRAM là 0x28. Trong 8 bit thì có 7 bit được định nghĩa và bit 6 không được định nghĩa. Nó chỉ có thể đọc và luôn có giá trị logic là 0.

Bit 7-ACD: Analog comparator disable

Đây là bit điều khiển. Bit này trực tiếp điều khiển hoạt động của AC (bộ so sánh tương tự). Nếu như bit này được set lên 1 thì nguồn cung cấp cho AC hoạt động bị tắt (turn off) và đồng nghĩa với việc nó không hoạt động. Và nếu nó được xóa thì AC được cấp nguồn và hoạt động bình thường. Chú ý: Ta có thể thay đổi giá trị logic của bit này lúc nào cũng được để ngưng hoạt động của chúng hoặc cho chúng hoạt động trở lại nhưng khi thay đổi giá trị logic của nó thì ngắt (ngắt của AC) cần bị cấm nếu không nó sẽ sinh ra một ngắt (Cụ thể là bit ACIE cần bị xóa).

Bit 5-ACO: Analog comparator output

Đây là bit trạng thái. Bit này được nối trực tiếp với đầu ra của bộ so sánh tương tự.

Bit 4-ACI: Analog comparator interrupt flag

Đây là bit trạng thái. Cờ báo ngắt của bộ so sánh tương tự. Nếu như cờ này được set và các ngắt được phép thì một chương trình phục vụ ngắt được gọi và chúng được xóa bằng phần cứng khi chương trình báo ngắt được phục vụ. Các trường

hợp làm thay đổi trạng thái cờ này ngoài việc thay đổi bit ACD sẽ được nói tới trong các bit 0 và 1.

Bit 3-ACIE:AC interrupt enable

Đây là bit điều khiển. Nếu bit này được set thì ngắt này được phép và ngược lại.

Bit 2-ACIC:Analog comparator input Capture Enable

Đây là bit điều khiển. Khi bit này được set lên 1 thì đầu ra của AC được nối trực tiếp vào đầu vào của chức năng bắt sự kiện của Timer/counter 1.(Đọc thêm timer/counter1).

Bit ACIS1 và ACIS0 :Ac interrupt mode select –Đây là hai bit điều khiển.

ACIS1	ACIS0	Chế độ ngắt
0	0	Theo mức
0	1	Dành riêng(chưa dùng đến)
1	0	Sườn xuống
1	1	Sườn lên

Hình 2.36 :Chế độ ngắt 2 bit ACIS1 và ACIS2

Chú ý: Các bit này cũng có thể được thay đổi bất cứ khi nào. Nhưng khi thay đổi thì ngắt của nó phải bị cấm.

Ta có thể sử dụng lệnh SBI hoặc CBIU để thay đổi trạng thái các bit trên thanh ghi này trừ bit ACI. Bit này sau khi được đọc cũng sẽ bị xóa (nếu nó được set).

Thiết lập port đầu vào cho bộ so sánh tương tự:

Hai chân PB2 và PB3 này cần được thiết lập là đầu vào bỏ điện trở treo.

Để lập trình cho AC ta bắt đầu các bước sau:

Bước 1: Thiết lập các chân đầu vào cho AC.

Bước 2: Chọn các chế độ cho AC ví như dùng ngắt ...

Bước 3: Khởi động AC bằng cách xóa bit ACD

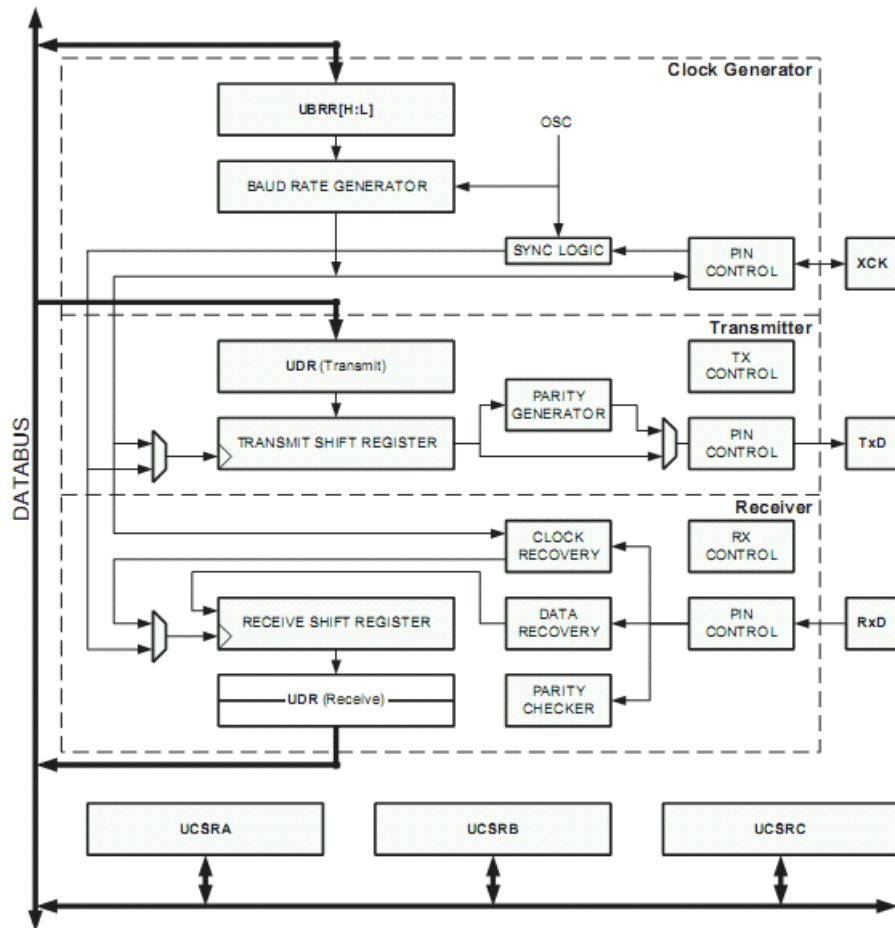
2.1. 11.USART (Universal asynchronous receiver/transmitter)

2.1.11.1 Đặc điểm

Bộ truyền nhận nối tiếp đồng bộ và bất đồng bộ là một thiết truyền thông nối tiếp có các chức năng chính như sau:

- Hoạt động song công (các thanh ghi truyền và nhận nối tiếp độc lập với nhau).
- Hoạt động đồng bộ hoặc bất đồng bộ
- Bộ tạo tốc độ baud có độ chính xác cao
- Hỗ trợ khung truyền nối tiếp với 5, 6, 7, 8, hoặc 9 bit dữ liệu và 1 hoặc 2 bit stop
- Kiểm tra chẵn lẻ
- Phát hiện tràn dữ liệu
- Phát hiện lỗi khung
- Lọc nhiễu, bao gồm phát hiện bit start lỗi và bộ lọc thông thấp số
- Ngắt khi kết thúc truyền, thanh ghi truyền hết dữ liệu và kết thúc nhận
- Chế độ truyền thông đa vi xử lý
- Chế độ truyền đồng bộ tốc độ cao

Sơ đồ khối của bộ USART như sau:



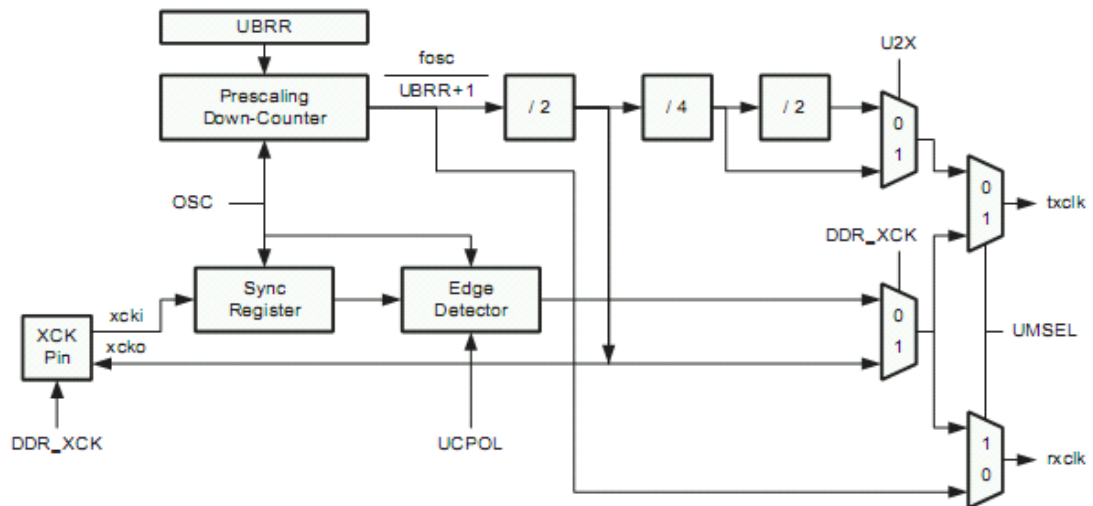
Hình 2.37: Sơ đồ khối bộ USART

USART bao gồm 3 phần chính: bộ tạo xung clock, bộ truyền và bộ nhận. Các thanh ghi điều khiển được sử dụng chung giữa các phần này.

2.1.11.2. Tạo xung clock

Bộ tạo xung clock tạo ra xung đồng hồ căn bản cho bộ truyền và bộ nhận. USART hỗ trợ 4 chế độ hoạt động xung clock: bất đồng bộ, bất đồng bộ tốc độ cao, truyền đồng bộ master và truyền đồng bộ slave.

Sơ đồ khối của bộ tạo xung clock như sau:



Hình 2.38: Đơn vị tạo xung clock

txclk: xung đồng hồ bộ truyền

rxclk: xung đồng hồ bộ nhận

xcki: tín hiệu vào từ chân XCK, sử dụng cho hoạt động truyền đồng bộ master

xcko: tín hiệu xung clock ngõ ra tới chân XCK, sử dụng cho hoạt động truyền đồng bộ slave

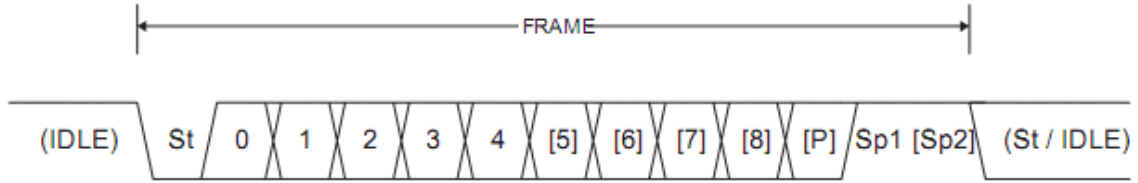
fosc: tần số từ chân XTAL

2.1.11.3. Định dạng khung truyền

USART chấp nhận tất cả 30 tổ hợp của các định dạng khung truyền sau đây:

- 1 bit start
- 5, 6, 7, 8, hoặc 9 bit dữ liệu
- Có hoặc không có bit chẵn lẻ
- 1 hoặc 2 bit stop

Một khung truyền bắt đầu với một bit start, theo sau đó là bit có trọng số thấp nhất (LSB) của dữ liệu (có thể lên tới 9 bit), kết thúc bằng bit có trọng số lớn nhất (MSB) và bit stop.



Hình 2.39: Định dạng khung truyền

St: bit start (mức thấp)

(n): bit dữ liệu (0 đến 8)

P: bit chẵn lẻ

Sp: bit stop (mức cao)

IDLE: không có dữ liệu truyền (mức cao trong suốt thời gian idle)

2.1.12. HỆ THỐNG XUNG CLOCK

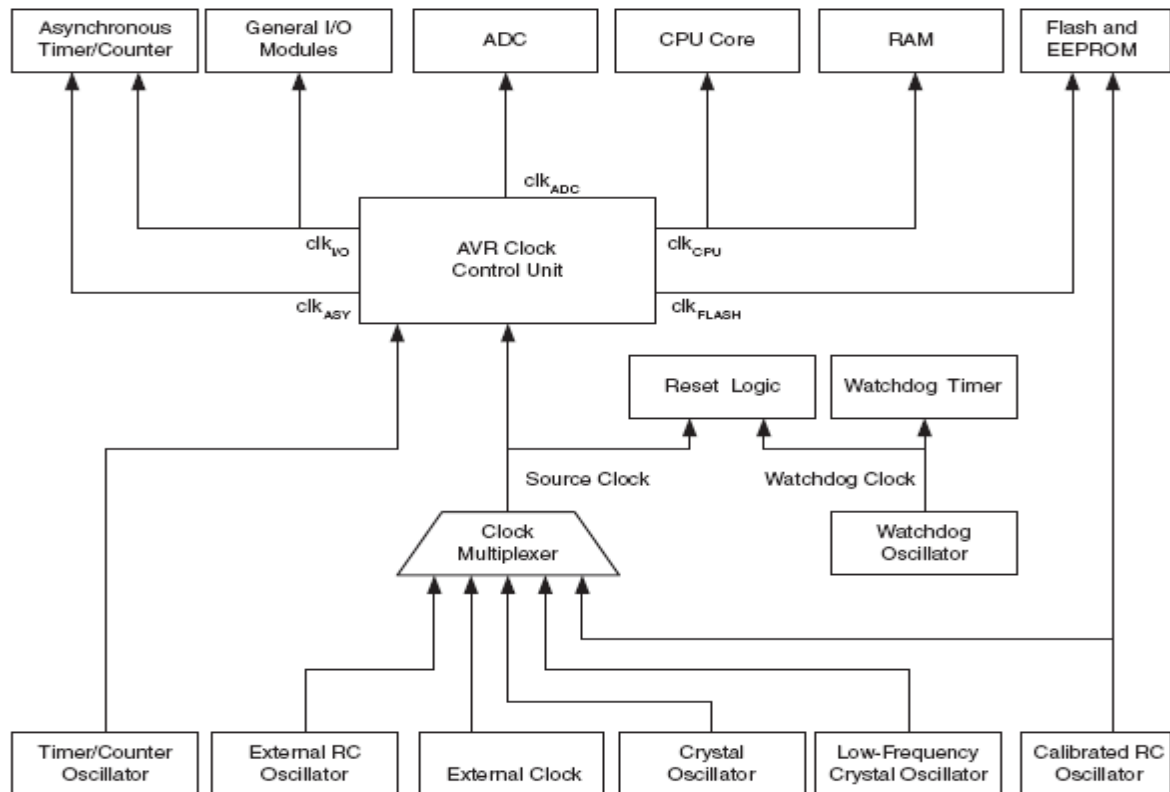
Để cấu hình cho chip hoạt động theo chế độ xung clock nào, người ta dùng các bit cầu chì (fuse bit) CKSEL 3, CKSEL2, CKSEL 1.

Ngoài ra khi vi điều khiển được đánh thức từ các chế độ nghỉ sang chế độ hoạt động bình thường, bộ tạo dao động cần có một khoảng thời gian để ổn định, khoảng thời gian này gọi là thời gian khởi động (start-up time). CPU chỉ thực hiện lệnh khi hết khoảng thời gian khởi động này.

Khi ta reset CPU cũng cần một khoảng thời gian trì hoãn (delay time) để nguồn nuôi đạt mức ổn định trước khi thực bắt đầu thực thi lệnh. Người ta dùng các bit cầu chì CKSEL 0, SUT1, SUT0 để thiết lập thời gian khởi động và thời gian trì hoãn. Khoảng thời gian khởi động và thời gian trì hoãn được đo được bằng một đồng hồ riêng, đó là bộ dao động Watchdog. Tần số của bộ dao động Watchdog phụ thuộc vào điện thế nguồn nuôi và nhiệt độ môi trường. Ở $V_{cc} = 5V$ và nhiệt độ $25^{\circ}C$ thì tần số của bộ dao động Watchdog là 1 MHz.

Liên quan đến việc thiết lập của hệ thống xung clock người ta còn dùng tới bit cầu chì CKOPT mà vai trò của nó khá linh hoạt tùy theo việc thiết lập xung clock cho hệ thống như thế nào. Hình 18 cho thấy ATmega128 có tới 7 bộ tạo xung clock có thể được lựa chọn.

Dưới đây là mô tả cụ thể cho từng trường hợp cấu hình xung clock của hệ thống.



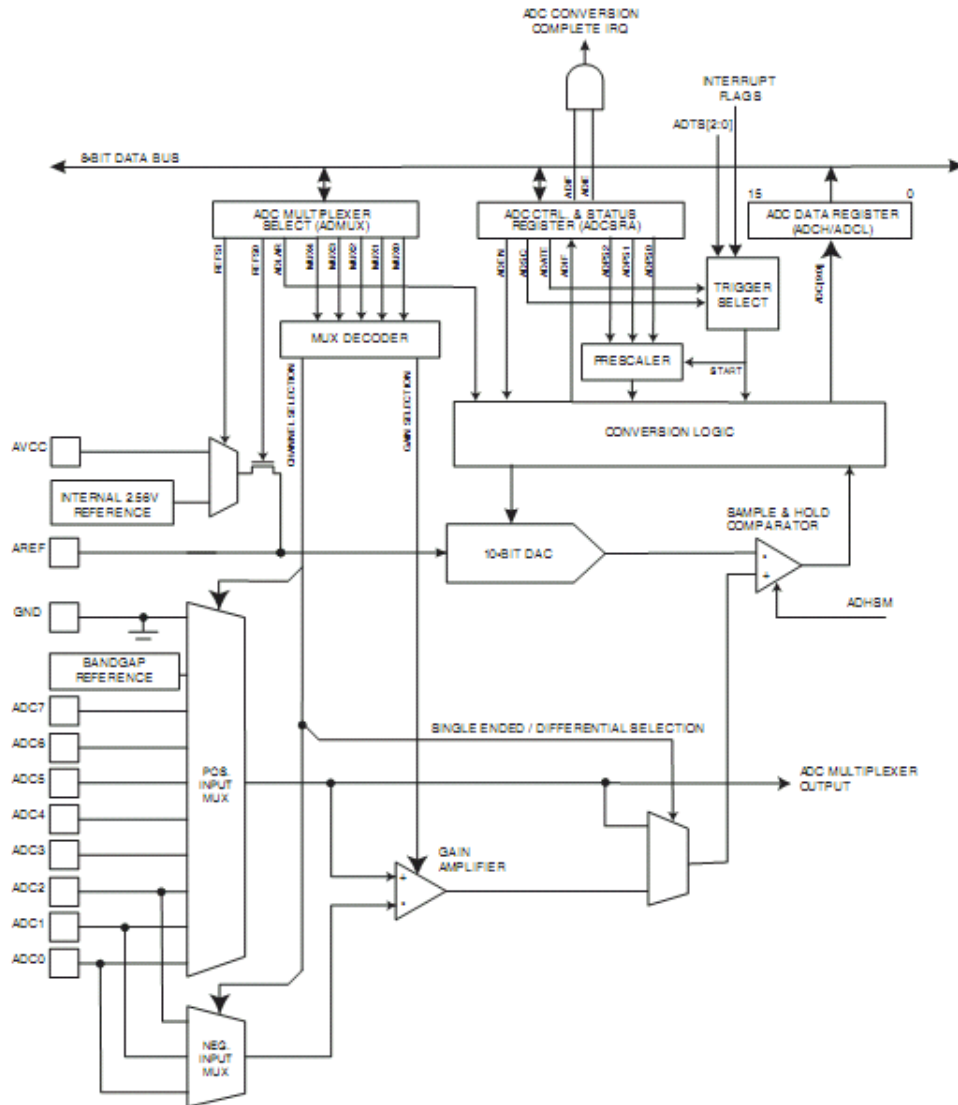
Hình 2.40: Hệ thống xung clock

2.1.13. BỘ BIẾN ĐỔI A/D(ANALOG/DIGITAL)

Vi điều khiển Atmega8 có một bộ biến đổi ADC tích hợp trong chip với các đặc điểm:

- Độ phân giải 10 bit
- Sai số tuyến tính: 0.5LSB

- Độ chính xác $\pm 2\text{LSB}$
- Thời gian chuyển đổi: $65\text{-}260\mu\text{s}$
- 6 Kênh đầu vào có thể được lựa chọn
- Có hai chế độ chuyển đổi free running và single conversion
- Có nguồn báo ngắt khi hoàn thành chuyển đổi
- Loại bỏ nhiễu trong chế độ ngủ



Hình 2.41: Sơ đồ bộ biến đổi A/D

Tám đầu vào của ADC là tám chân của PORTA và chúng được chọn thông qua một MUX.

Để điều khiển hoạt động vào ra dữ liệu của ADC và CPU chúng ta có 3 thanh ghi: ADMUX là thanh ghi điều khiển lựa chọn kênh đầu vào cho ADC, ADCSRA là thanh ghi điều khiển và thanh ghi trạng thái của ADC, ADCH và ADCL là 2 thanh ghi dữ liệu.

Nguyên tắc hoạt động và lập trình điều khiển

ADC có nhiệm vụ chuyển đổi tín hiệu điện áp tương tự thành tín hiệu số có độ phân giải 10 bit. Với giá trị nhỏ nhất của điện áp đặt ở chân AGND và giá trị cực đại của điện áp tương tự được mắc vào chân AREF. Tám kênh tương tự đầu vào được chọn lựa thông qua ADMUX và ADMUX này được điều khiển bởi thanh ghi ADMUX.

ADC này có thể hoạt động được ở hai chế độ. Đó là chuyển đổi đơn: chỉ chuyển đổi một lần khi có lệnh chuyển đổi và chế độ tự chuyển đổi (Free running mode) đây là chế độ mà ADC tự động chuyển đổi khi được hoạt động và công việc chuyển đổi có tính tuần hoàn (chỉ cần khởi động một lần).

ADC được phép hoạt động nhờ thiết lập bit ADEN. Quá trình chuyển đổi được bắt đầu bằng việc ghi vào bit ADSC mức logic 1 và trong suốt quá trình chuyển đổi bit này luôn được giữ ở mức cao. Khi quá trình chuyển đổi hoàn thành thì bit này được xóa bằng phần cứng và cờ AIDF được bật lên.

Dữ liệu sau khi chuyển đổi được đưa ra thanh ghi dữ liệu ADCL và ADCH, nhưng chú ý khi đọc dữ liệu từ hai thanh ghi này thì đọc ADCL trước rồi mới đọc ADCH. Nếu đọc ADCH trước thì dữ liệu cập nhật có thể ghi đè lên ADCL (Vi điều khiển nghĩ rằng đã đọc xong dữ liệu).

Để điều khiển vào ra dữ liệu với ADC, các bước thực hiện như sau:

Bước 1: Định nghĩa các cổng vào cho tín hiệu tương tự

Xóa bit tương ứng với chân đó trong thanh ghi DDRA. Sau đó loại bỏ điện trở treo bằng cách xóa bit tương ứng ở thanh ghi PORTA.

Bước 2: Chọn kênh tương tự vào (chọn chân vào cho ADC) thông qua thanh ghi ADMUX (có thể thay đổi trong quá trình hoạt động).

Bước 3: Thiết lập các thông số cho ADC

Tốc độ chuyển đổi thông qua xung nhịp chuyển đổi.

Chế độ chuyển đổi : đơn hoặc tự động.

Sử dụng ngắt hoặc không.

Bước 4: Bắt đầu chuyển đổi và đọc dữ liệu.

2.2. CÁC LINH KIỆN KHÁC

2.2.1. Motor một chiều

Sử dụng 3 motor 12V DC làm động cơ truyền động chính cho băng tải.



Hình 2.42: Motor một chiều 12V

Các thông số của motor:

Điện áp cấp nguồn: 12V DC, 2A

Công suất: 25W

Tốc độ tối đa 200vòng/phút

2.2.2. Biến áp cấp nguồn

Để cấp nguồn cho mạch động lực và mạch điều khiển trong mô hình em sử dụng 1 biến áp hạ áp.



Hình 2.43: Biến áp cấp nguồn

Các thông số của biến áp:

- Nguồn cấp vào biến áp: 220 VAC
- Nguồn ra 6V, 9V, 12V, 15V, 18V, 24 VAC
- Dòng định mức: 5A

2.2.3. LCD

Hiển thị kết quả đo được, giao diện cài đặt thân thiện với người sử dụng.



Hình 2.44: LCD hiển thị

Thông số LCD C1602A:

COB 16 x 2 dòng, STN, đèn nền White LED, Wide temp., IC:
SPLC780D

Kích thước :80.0W x 36.0H x 13.5T mm

Viewing area:64.5W x 14.5H mm

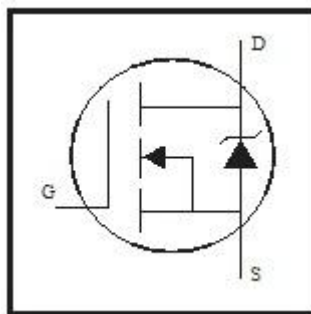
2.2.4. Mosfet

Là van động lực dùng để điều khiển đóng mở cấp nguồn cho dây mayso



Hình 2.45: Mosfet IRFZ44N

CẤU TẠO



Hình 2.46: Cấu tạo mosfet IRFZ44N

Thông số của Mosfet

Giải thích nguyên lý hoạt động

Dùng biến áp 5A để chuyển từ nguồn 220 VAC sang nguồn 18 VAC

Nguồn 18 VAC này sẽ cho qua cầu diode để nắn dòng xoay chiều thành 1 chiều.
Có điện áp khoảng 24 VDC

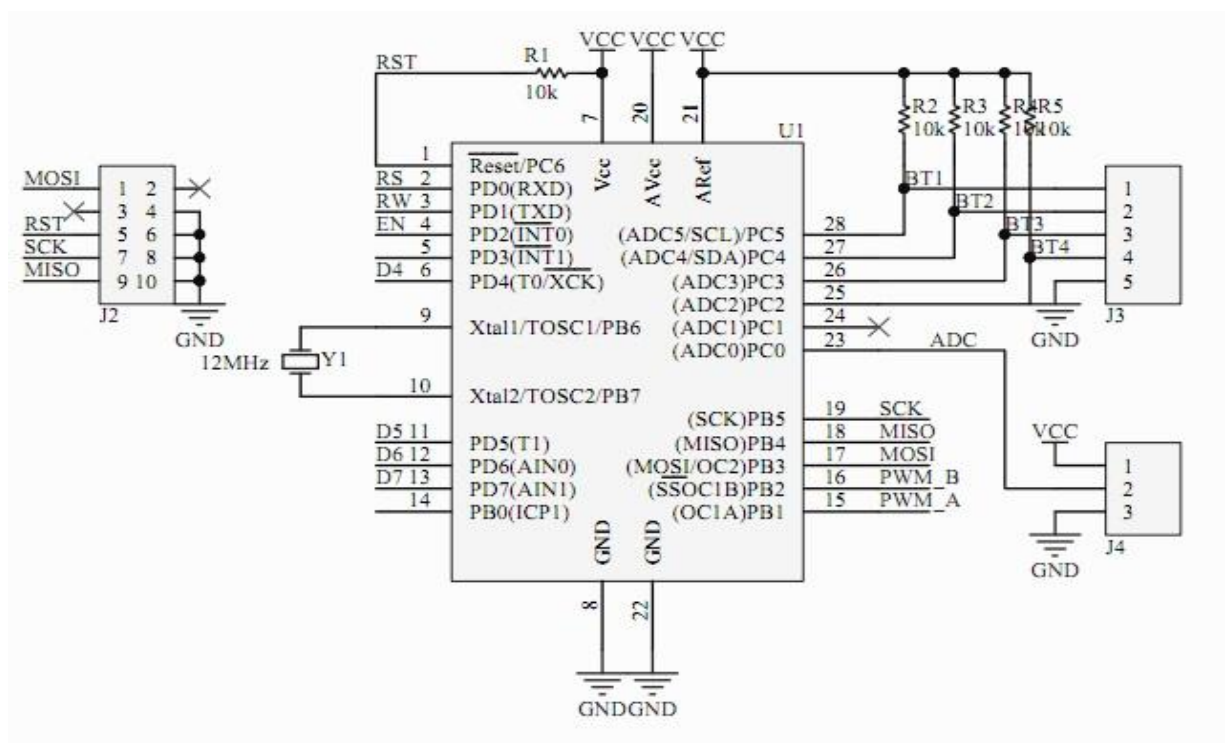
Diode D1 là led để báo có nguồn cấp cho mạch

Các tụ trong mạch có tác dụng lọc nhiễu nguồn đầu vào cũng như lọc nhiễu nguồn đầu ra. Làm cho nguồn tương đối ổn định, không bị ảnh hưởng nhiều bởi tải (tải nhỏ).

IC 7805 có tác dụng ổn định điện áp đầu ra là 12VDC.

3.2.2. Khối mạch điều khiển ATmega8

Sơ đồ nguyên lý

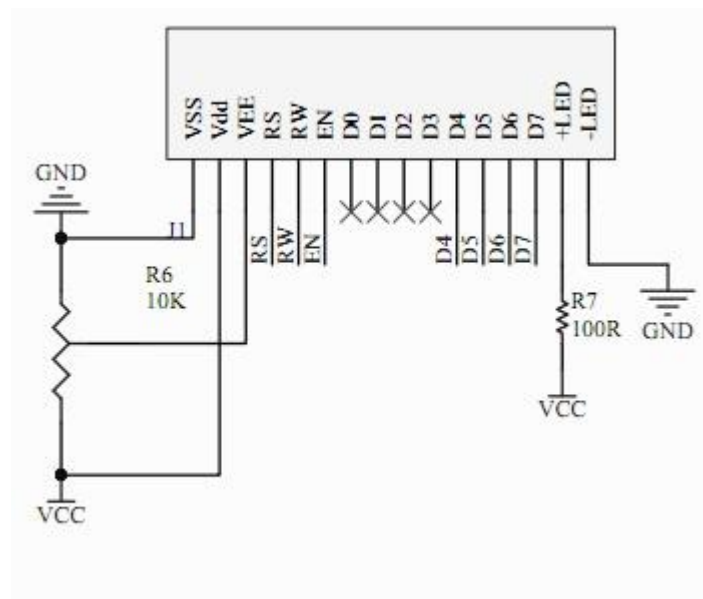


Hình 3.3: Mạch điều khiển cho ATmega8

Giải thích nguyên lý hoạt động:

- Sử dụng các cổng kết nối J2, J3, J4. Trong đó J2 để nạp chương trình điều khiển cho ATmega8, J3 thiết lập các chế độ do người sử dụng vận hành.
- Chân PB1, PB2 tạo tín hiệu điều khiển Mosfet
- Các chân PD0, PD1, PD2, PD4, PD5, PD6, PD7 tạo tín hiệu đưa ra LCD
- Chân PC0 nhận tín hiệu phản hồi từ cảm biến nhiệt
- Các chân GND nối mass

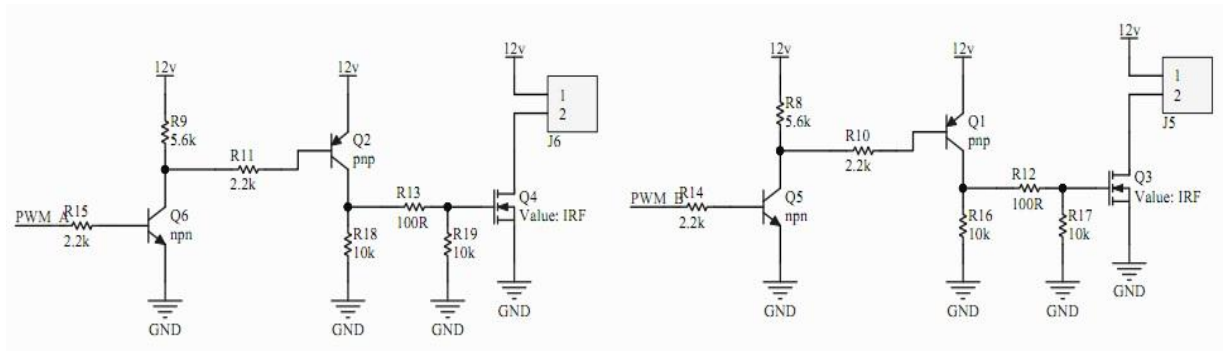
3.2.3. Khối mạch điều khiển LCD



Hình 3.4: mạch điều khiển LCD

Là mạch tích hợp sẵn có các chân RS, RW, EN, D4, D5, D6, D7 nhận tín hiệu điều khiển từ ATmega8.

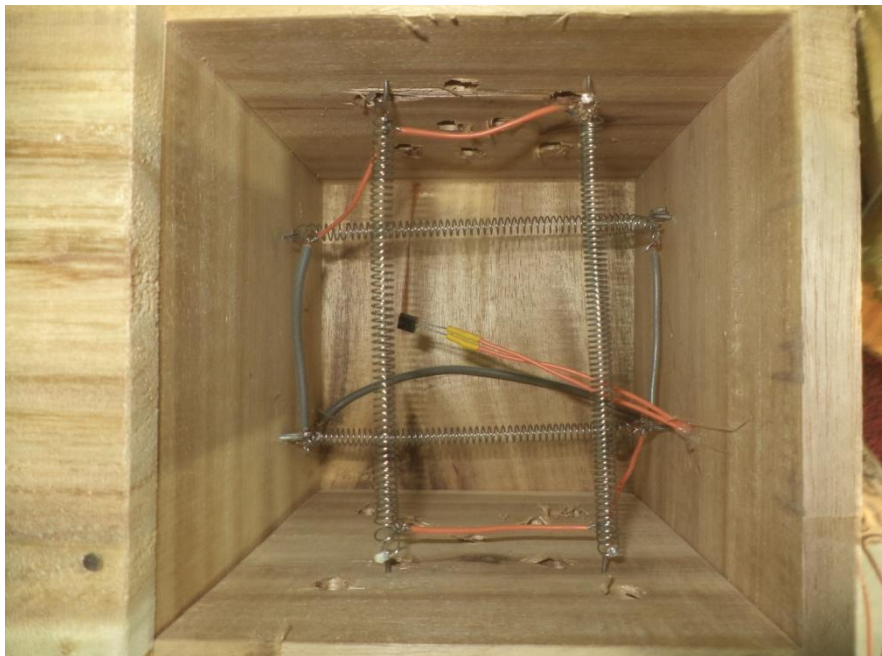
3.2.4. Khối mạch điều khiển Mosfet



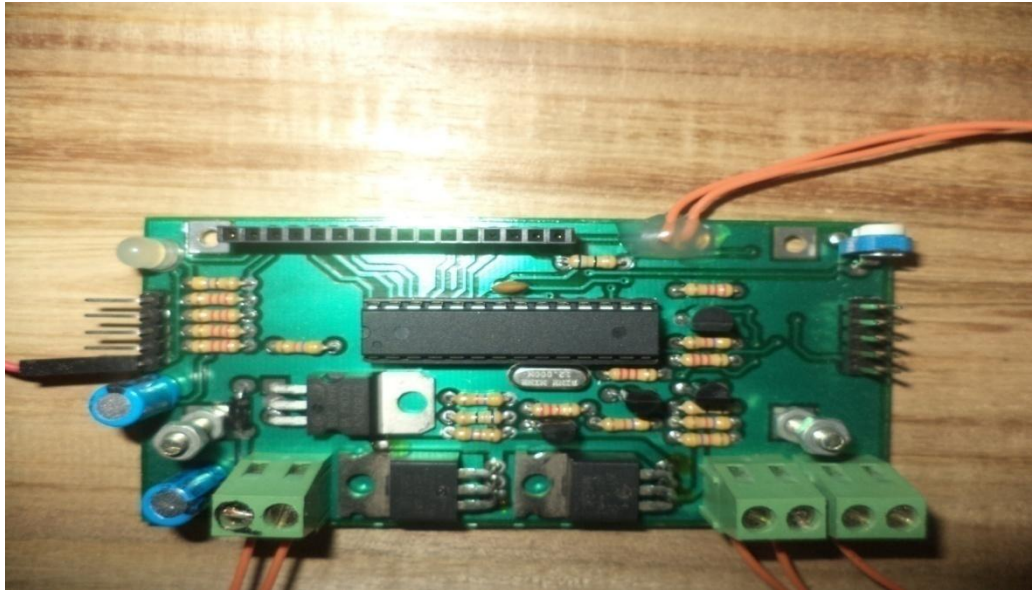
Hình 3.5: Khối mạch điều khiển Mosfet

Một số hình ảnh về mô hình

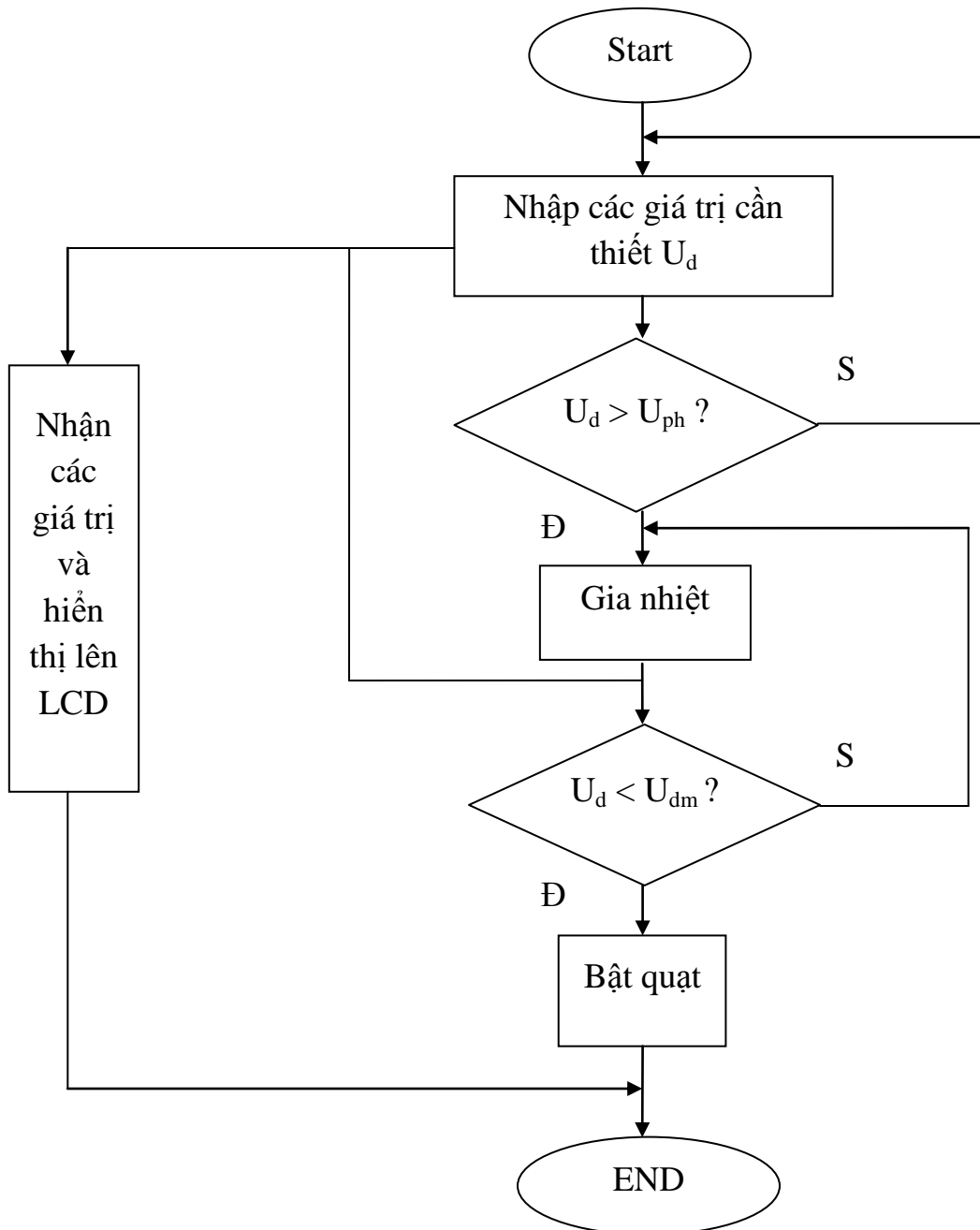
a. Hộp gia nhiệt



b. Mạch điều khiển



3.3 LƯU ĐỒ THUẬT TOÁN ĐIỀU KHIỂN



Hình 3.7: Lưu đồ thuật toán chương trình điều khiển

3.4. CHƯƠNG TRÌNH ĐIỀU KHIỂN

```
#include <mega8.h>
```

```
#include <delay.h>
```

```
unsigned char Data_LM35=0, ev_1=0, ev_2=0, value_tep=0, t=0;
```

```
// Alphanumeric LCD functions
```

```
#include <alcd.h>
```

```
// Timer 0 overflow interrupt service routine
```

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
```

```
// Reinitialize Timer 0 value
```

```
TCNT0=0x08;
```

```
// Place your code here
```

```
    t++;
```

```

if(t==70 && run==1)//Neu dat 20ms va dc nap toc do
{
    t=0;

    Data_LM35=((adc_data[0]-558.558)/2.048);//5v

    fb=Data_LM35;// Sao chep

    ev_1=value_tep-fb;// Sai lech hien tai

    PWM=PWM+0.9*ev_1+0.001*(ev_1+ev_2);// Tinh toan PWM

    ev_2=ev_1;// Sai lech 2

    Display=1;
}

else if(t==70 && run==0)

{
    t=0;
}

}

#define FIRST_ADC_INPUT 0

#define LAST_ADC_INPUT 0

```

```

unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];

#define ADC_VREF_TYPE 0x00

// ADC interrupt service routine

// with auto input scanning

interrupt [ADC_INT] void adc_isr(void)
{
    static unsigned char input_index=0;

    // Read the AD conversion result

    adc_data[input_index]=ADCW;

    // Select next ADC input

    if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
        input_index=0;

    ADMUX=(FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff))+input_index;

    // Delay needed for the stabilization of the ADC input voltage

    delay_us(10);

    // Start the AD conversion

    ADCSRA|=0x40;
}

```

```

// Declare your global variables here

void lcd_put_int(int num)//Xuat 1 so nguyen ra LCD
{
    int temp;

    unsigned char i = 0, c[5];

    temp = num;

    if (temp != 0) {
        if (temp < 0){
            lcd_putchar('-');

            temp = - temp;
        }

        while(temp){

            c[i++] = temp%10;

            temp /= 10;

        }

        while(i) lcd_putchar(c[--i] + '0');
    }
}

```

```
    else lcd_putchar('0');  
}  
  
void main(void)  
{  
    // Declare your local variables here  
  
    // Input/Output Ports initialization  
    // Port B initialization  
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out  
    Func0=In  
    // State7=T State6=T State5=T State4=T State3=T State2=0 State1=0 State0=T  
    PORTB=0x00;  
    DDRB=0x06;  
  
    // Port C initialization  
    // Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTC=0x00;

DDRC=0x00;

// Port D initialization

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTD=0x00;

DDRD=0x00;

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: 11.719 kHz

TCCR0=0x05;

TCNT0=0x08;

// Timer/Counter 1 initialization

// Clock source: System Clock
```

```
// Clock value: 46.875 kHz

// Mode: Ph. correct PWM top=0x00FF

// OC1A output: Non-Inv.

// OC1B output: Non-Inv.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0xA1;

TCCR1B=0x04;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;
```



```
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer2 Stopped
```

```
// Mode: Normal top=0xFF
```

```
// OC2 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR2=0x00;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
MCUCR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x01;
```

```
// USART initialization

// USART disabled

UCSRB=0x00;

// Analog Comparator initialization

// Analog Comparator: Off

// Analog Comparator Input Capture by Timer/Counter 1: Off

ACSR=0x80;

SFIOR=0x00;

// ADC initialization

// ADC Clock frequency: 187.500 kHz

// ADC Voltage Reference: AREF pin

ADMUX=FIRST_ADC_INPUT | (ADC_VREF_TYPE & 0xff);

ADCSRA=0xCE;

// SPI initialization

// SPI disabled
```

```
SPCR=0x00;

// TWI initialization

// TWI disabled

TWCR=0x00;

// Alphanumeric LCD initialization

// Connections are specified in the

// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:

// RS - PORTD Bit 0

// RD - PORTD Bit 1

// EN - PORTD Bit 2

// D4 - PORTD Bit 4

// D5 - PORTD Bit 5

// D6 - PORTD Bit 6

// D7 - PORTD Bit 7

// Characters/line: 16

lcd_init(16);
```

```
// Global enable interrupts

#asm("sei")

while (1)
{
    // Place your code here

    if(Display==1)
    {
        Display=0;

        lcd_clear();// Xoa LCD

        lcd_putsf("NHIET DO: ");

        lcd_put_int(Data_LM35);
    }
}
}
```

KẾT LUẬN

Sau ba tháng làm tốt nghiệp, được sự giúp đỡ tận tình của GS. TSKH Thân Ngọc Hoàn cùng với sự nỗ lực của bản thân em đã hoàn thành đồ án tốt nghiệp với đề tài:

“ Trang bị điện máy sản xuất thức ăn chăn nuôi, đi sâu thiết kế hệ thống sấy khô”

Trong quá trình làm đồ án, về cơ bản em đã hệ thống được kiến thức, tìm hiểu được một số ứng dụng của Vi điều khiển ATmega8 (họ Vi điều khiển tương đối mạnh và được ứng dụng nhiều), hệ thống gia nhiệt...

Do thời gian làm đồ án và kiến thức bản thân còn hạn chế vì vậy bản đồ án còn nhiều thiếu sót, mới chỉ dừng lại ở mức độ tổng quát và thiết kế tổng quát mạch phần cứng. Để đề tài được hoàn thiện hơn, một lần nữa em mong nhận được sự đóng góp ý kiến, chỉ bảo của các thầy cô và các bạn. Cũng như có kiến thức nhất định sau khi ra trường.

Em xin chân thành cảm ơn !

Hải Phòng, ngày....., thángnăm.....

Sinh viên thực hiện

Bùi Tiến Tùng

TÀI LIỆU THAM KHẢO

1. Hồ Trung Mỹ (2007) Vi xử lý. Nhà xuất bản đại học Quốc Gia
2. Datasheet ATmega8
3. Trương Sa Sanh (chủ biên),(2003) . Kỹ thuật điện đại cương. Nhà xuất bản đại học Quốc Gia
4. Phạm Công Ngô (1985) Lý thuyết điều khiển tự động. Nhà xuất bản Khoa học và Kỹ Thuật
5. Các tài liệu về Điện Tử Công Suất
6. Các tài liệu từ internet, và các đề án của các anh chị khóa trên.

MỤC LỤC

Lời mở đầu	1
CHƯƠNG 1. GIỚI THIỆU VỀ QUY TRÌNH SẢN XUẤT THỨC ĂN CHĂN NUÔI	2
1.1. GIỚI THIỆU CHUNG VỀ TÌNH HÌNH SẢN XUẤT THỨC ĂN CHĂN NUÔI	2
1.1.1. Sơ lược về tình hình sản xuất thức ăn viên ở nước ta	2
1.1.2. Các nguồn nguyên liệu thường dùng làm thức ăn cho gia súc , gia cầm, thủy cầm (GSTC)	3
1.2. PHƯƠNG PHÁP, QUY TRÌNH SẢN XUẤT THỨC ĂN CHĂN NUÔI	4
1.2.1. Công nghệ sản xuất thức ăn cho GSTC	4
1.2.2. Đặc điểm và yêu cầu kích thước của thức ăn chăn nuôi	5
1.2.3. Phương pháp xác định độ nhỏ bột nghiền	6
1.2.4. Phương pháp xác định độ trộn đều bột nghiền	8
1.2.5. Phương pháp xác định độ bền và độ cứng viên thức ăn	9
1.2.6. Sơ bộ các công đoạn sản xuất thức ăn chăn nuôi	11
1.2.6.1. Công đoạn nghiền	12
1.2.6.2. Công đoạn định lượng	13
1.2.6.3. Công đoạn trộn	13

1.2.6.4. Công đoạn vận chuyển và bộ phận trung gian	14
1.2.6.5. Công đoạn ép viên	14
1.2.6.6. Công đoạn sấy và làm nguội	17
Chương 2.	
CÁC PHẦN TỬ SỬ DỤNG TRONG HỆ THỐNG SẤY BĂNG TẢI	18
2.1. GIỚI THIỆU VỀ VI ĐIỀU KHIỂN ATMEGA8	18
2.1.1. Tổng quan về vi điều khiển ATMEGA8	18
2.1.2. Cấu trúc chung AVR	24
2.1.2.1. Cấu trúc tổng quát	24
2.1.2. Cấu trúc chung AVR	25
2.1.2.1. Cấu trúc tổng quát	25
2.1.2.4. Các thanh ghi chức năng chung	26
2.1.2.5. Con trỏ ngăn xếp (SP)	27
2.1.3. Cấu trúc ngắt của ATMEGA8	27
2.1.3.1. Khái niệm về ngắt	27
2.1.3.2. Trình phục vụ ngắt và bảng vector ngắt	28
2.1.3.3. Bảng vector ngắt	29
2.1.3.4. Thứ tự ưu tiên ngắt	29
2.1.3.5. Ngắt trong ngắt	30
2.1.4. Cấu trúc bộ nhớ	31

2.1.4.1. Bộ nhớ chương trình (Bộ nhớ Flash)	31
2.1.4.2. Bộ nhớ dữ liệu SRAM	32
2.1.4.3. Bộ nhớ dữ liệu EEPROM	32
2.1.5. CÁC CÔNG VÀO RA (I/O)	33
2.1.5.1. Các chức năng của Port B	33
2.1.5.2. Các chức năng của Port C	35
2.1.5.3. Các chức năng của Port D	36
2.1.5.4. Mô tả thanh ghi của port I/O	37
2.1.6. BỘ ĐỊNH THỜI 8BIT TIMER/COUNTER 0	40
2.1.6.1. Hoạt động của bộ Timer/Couter	41
2.1.6.2. Đơn vị đếm	43
2.1.7. Bộ định thời/đếm TIMER/COUNTER 1 16-BIT	44
2.1.7.1. Sơ đồ khối và một số đặc điểm	44
2.1.7.2. Một số định nghĩa	45
2.1.8. SPI(SERIAL PERIPHERAL INTERFACE)	45
2.1.9. TWI – TWO WIRE INTERFACE	46
2.1.10. Bộ so sánh tương tự (ALALOG COMPARATOR)	47
2.1. 11.USART (Universal asynchronous receiver/transmitter)	50
2.1.11.1 Đặc điểm	50
2.1.11.2. Tạo xung clock	51

2.1.12. HỆ THỐNG XUNG CLOCK	53
2.1.13. BỘ BIẾN ĐỔI A/D(ANALOG/DIGITAL)	54
2.2. CÁC LINH KIỆN KHÁC	57
2.2.1. Motor một chiều	57
2.2.2. Biến áp cấp nguồn	58
2.2.3. LCD	58
2.2.4. Mosfet	58
Chương 3.	
THIẾT KẾ VÀ THI CÔNG HỆ THỐNG SẤY KHÔ	60
3.1 SƠ ĐỒ NGUYÊN LÝ	60
3.2. THIẾT KẾ MẠCH ĐIỆN	60
3.2.1. Mạch nguồn 12V DC	60
3.2.2. Khối mạch điều khiển ATmega8	61
3.2.3. Khối mạch điều khiển LCD	62
3.2.4. Khối mạch điều khiển Mosfet	63
3.3 LƯU ĐỒ THUẬT TOÁN ĐIỀU KHIỂN	65
3.4. CHƯƠNG TRÌNH ĐIỀU KHIỂN	66
KẾT LUẬN	77
TÀI LIỆU THAM KHẢO	78
MỤC LỤC	79