**BAB II**

**ANALISIS DAN PERANCANGAN**

Dalam sebuah sistem pengendalian perangkat keras diperlukan adanya sebuah komputer sebagai pengolah data sesuai dengan program yang bertugas untuk menjalankan perangkat yang dirancang, sekaligus untuk menampilkan hasil dari perangkat yang dikerjakan sesuai dengan yang diinginkan.

* 1. **ORGANISASI KOMPUTER**

Secara umum sebuah sistem komputer terdiri dari tiga bagian utama, yaitu CPU (*Central Processing Unit*), memori dan piranti *input/output* (*I/O*). Ketiga bagian utama ini satu dengan yang lain dihubungkan dengan tiga jalur utama yang berupa bus. Tiga bus tersebut adalah bus alamat, bus data dan bus kendali. Gambar 2.1 menunjukkan diagram blok sistem komputer secara umum.

CPU merupakan bagian terpenting dalam sistem komputer, karena CPU mengatur aktivitas pengolahan data serta pengaturan piranti-piranti lain yang terdapat dalam sistem komputer. Tugas utama CPU antara lain yaitu :

1. Mengambil dan melaksanakan instruksi yang tersimpan dalam memori,
2. Mengambil dan menyimpan data dari dalam memori yang diolah pada pelaksanaan suatu instruksi,
3. Mengirim data ke pirantri I/O dan mengambil data dari piranti I/O
4. Mengendalikan piranti penunjang CPU guna pengaturan DMA dan pelayanan interupsi

Bus Data

**Piranti**

**Input**

Bus Kendali Bus Kendali

**Memori**

**CPU**

**I/O**

**Port**

**Piranti**

**Output**

Bus Alamat

**Gambar 2.1 Diagram blok sistem komputer**

Piranti pada komputer bertugas menyimpan data atau instruksi yang harus dieksekusi oleh CPU. Piranti memori terdiri atas dua macam, yaitu memori hanya baca (*read only memory*) yang biasa disebut ROM dan memori baca/tulis (*read/write memory*) yang biasa disebut RAM (*random access memory*).

Bagian input dan output berfungsi menjembatani sistem komputer dengan dunia luar. Bagian ini berupa piranti I/O, seperti layar monitor, papan ketik (*keyboard*), pencetak (*printer*) dan lain-lain.

Bus-bus yang menghubungkan CPU, memori dan terminal I/O dapat dibagi menjadi tiga bagian, yaitu :

1. **Bus data**

Bus data berfungsi sebagai jalur pertukaran data dari berbagai piranti subsistem, seperti pengiriman data dari CPU ke memori dan piranti I/O, pengambilan data dari memori dan piranti I/O ke CPU. Bus data bersifat dua arah (*bi-directional*) dan “tristate” (dapat bernalar nol, satu atau berimpedansi tinggi).

1. **Bus alamat**

Bus alamat berfungsi sebagai jalur untuk mengirimkan data alamat. Dengan jalur ini CPU mengirimkan atau mengeluarkan alamat lokasi memori yang dibaca atau ditulisi. Dengan jalur ini pula CPU mengirimkan alamat piranti I/O yang akan dibaca atau ditulis. Besarnya lokasi memori atau I/O yang dapat dialamati tergantung lebar bus alamat. Bus alamat juga bersifat “*tristate*”.

1. **Bus kendali**

Bus kendali berfungsi sebagai jalur untuk mengirimkan sinyal kendali ke suatu piranti. Beberapa sinyal kendali utama selalu ada dalam komputer yaitu sinyal baca memori, sinyal tulis memori, sinyal baca I/O dan sinyal tulis I/O, dan sinyal-sinyal kendali input ke prosesor. Bus kendali ini juga bersifat “*tristate*”.

* 1. **PORT PARALEL**

Pada komputer standar, terdapat sebuah port paralel yang digunakan untuk komunikasi antara komputer dengan printer, karena itu sering disebut juga disebut dengan port printer.

Nomor alamat port paralel tidak tetap, tetapi tergantung pada informasi yang terdapat pada low memori di lokasi 0040:0080H sampai 0040:0008FH. Namun demikian untuk sebagian besar komputer yang mengikuti standar IBM PC, maka nomor port paralel tersebut berada pada 378H, 379H dan 37AH.

* + 1. **Register Port Paralel**

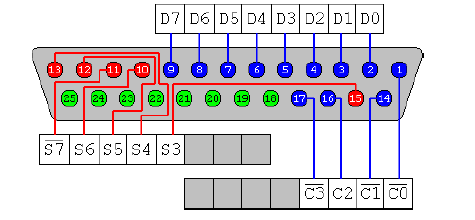
Port paralel khusus didesain untuk berhubungan dengan printer, namun ia juga dapat digunakan sebagai *interface* masukan dan keluraran umum, bagi peralatan ataupun program yang membutuhkan media masukan dan keluaran.

Port paralel memiliki 12 register keluaran yang dapat digunakan untuk menulis dan membaca pada alamatnya dengan menggunakan perintah keluaran dan masukan dari prosesor.

Port paralel juga mempunyai lima register masukan yang dapat dibaca dengan menggunakan perintah masukan dari prosesor.

Port paralel diakses melalui 8 pin keluaran data port pada alamat 378H, 5 pin masukan status port pada alamat 379H dan 4 pin keluaran control port pada alamat 37AH. Sehingga semuanya berjumlah 12 pin keluaran dan 5 pin masukan.

Data port – alamat 378H



Staus Port – alamat 379H

Control port – alamat 37AH

**Gambar 2.2 Tata Letak Register Port Paralel**

Bila diperhatikan pada gambar 2.2 pada pin S7 port register status dan pin C0, C1 dan C3 port register control, terdapat garis pada masing-masing pin. Hal ini menandakan bahwa pin tersebut diinvert (dibalik nilainya). Sehingga bila dikirim data 0, yang akan keluar adalah data 1, dan sebaliknya.

* + 1. **Pinout Konektor DB25 Female**

Konektror DB-25 adalah konektor yang paling banyak ditemukan pada port paralel komputer. Standar IEEE.1284 menetapkan 3 macam jenis konektor yang berbeda, yaitu :

1. 1284 Tipe A adalah konektor DB-25 yang banyak kita jumpai pada komputer-komputer saat ini;
2. 1284 Tipe B adalah konektor Centronics 34 pin yang banyak dijumpai pada pencetak; dan
3. 1284 Tipe C adalah konektor 36 pin yang mirip dengan Centronics, namun lebih kecil. Konektor ini diklaim memiliki pengunci (*latch*) jenis klip, sifat elektrik yang lebih baik serta mudah dirakit.

Berikut ini adalah spesifikasi pinout konektor DB-25 female.

**Tabel 2.1. Spesifikasi Pin Port Printer pada DB-25**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin | Nama Sinyal | Input / Output | Port | Membalik |
| 1 | Strobe | Output | Control Port | Ya |
| 2 | Data 0 | Input / Output | Data Port | Tidak |
| 3 | Data 1 | Input / Output | Data Port | Tidak |
| 4 | Data 2 | Input / Output | Data Port | Tidak |
| 5 | Data 3 | Input / Output | Data Port | Tidak |
| 6 | Data 4 | Input / Output | Data Port | Tidak |
| 7 | Data 5 | Input / Output | Data Port | Tidak |
| 8 | Data 6 | Input / Output | Data Port | Tidak |
| 9 | Data 7 | Input / Output | Data Port | Tidak |
| 10 | Acknowledge | Input | Status Port | Tidak |
| 11 | Busy | Input | Status Port | Ya |
| 12 | Paper Out | Input | Status Port | Tidak |
| 13 | Selected (Online) | Input | Status Port | Tidak |
| 14 | Auto Feed | Output | Control Port | Ya |
| 15 | Error | Input | Status Port | Tidak |
| 16 | Reset | Output | Control Port | Tidak |
| 17 | Select Input | Output | Control Port | Ya |
| 18–25 | Ground | - | - | - |

* 1. **ADC 0809**

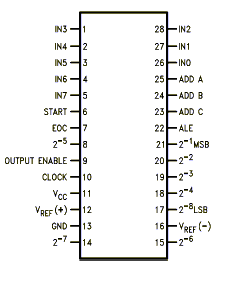
Fungsi utama dari sebuah ADC (*Analog to Digital Converter*) adalah pengubah sinyal atau tegangan analog menjadi data digital yang nantinya akan diproses lebih lanjut dengan komputer.

Perlu dicatat bahwa data-data digital yang dihasilkan ADC hanyalah merupakan pendekatan terhadap masukan analog. Hal ini karena tidak mungkin melakukan konversi secara sempurna berkaitan dengan kenyataan bahwa informasi digital berubah dalam *step-step*, sedangkan analog berubahnya secara kontinyu.

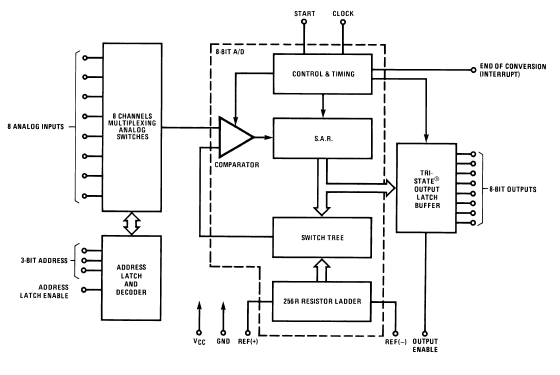
Misalnya ADC dengan resolusi 8 bit menghasilkan bilangan 0 sampai dengan 255 (256 bilangan dan 255 *step*), dengan demikian tidak mungkin menyajikan semua kemungkinan nilai-nilai analog. Jika sekarang resolusinya menjadi 20 bit maka akat terdapat 1.048.578 *step*, semakin banyak kemungkinan nilai-nilai analog yang bisa disajikan. Bagaimanapun pada sebuah *step* terdapat tak-terhingga kemungkinan nilai-nilai analog untuk sembarang ADC yang dapat diperoleh di dunia ini. Sehingga ada yang dibuat manusia (*human-made*) tidak akan pernah bisa menyamai kondisi dunia nyata.

ADC yang digunakan adalah ADC 0809 dikarenakan memiliki beberapa kelebihan dari ADC lainnya serta mudah diperoleh di pasaran. Garis besar spesifikasi ADC 0809 adalah sebagai berikut :

* Resolusi 8 bit.
* Masukan analog ada 8 kanal, dipilih dengan teknik multiplekser.
* Keluaran (8 bit).
* Catu daya 5 volt, 3mA
* Rasiometrik terhadap tegangan acuan.
* Waktu konversi 100 mikrodetik.
* Frekuensi detak kerja (*clock*) 10 s/d 1200 KHZ.



**Gambar 2.3 Konfigurasi pin ADC 0809**



**Gambar 2.4 Blok diagram ADC 0809 / 0808**

ADC ini dirancang untuk dapat bekerja bersama dengan mikroprosesor. Sinyal kontrolnya cocok dengan level TTL (*Transistor Transistor Logic*), perhatikan gambar 2.3 unturk daftar pin-pin ADC 0809 dan gambar 2.4 untuk blok diagram dari ADC ini.

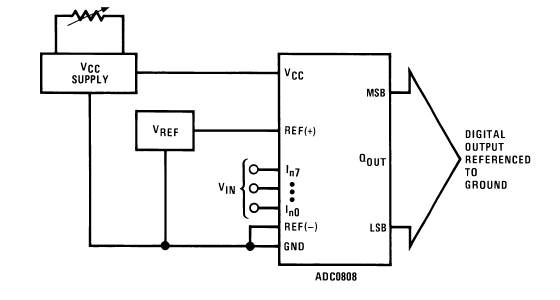
ADC 0809 menggunakan teknik multiplekser untuk pemilihan 8 input analognya, oleh karena itu perlu diperhatikan pemberian alamat input untuk menentukan input analog yang dipilih. Perhatikan tabel 2.2 berikut :

**Tabel 2.2 Pemilihan Input ADC 0809**

|  |  |  |  |
| --- | --- | --- | --- |
| **Logika** | | | **Keterangan** |
| **Add 2** | **Add 1** | **Add 0** |
| 0 | 0 | 0 | Input 0 |
| 0 | 0 | 1 | Input 1 |
| 0 | 1 | 0 | Input 2 |
| 0 | 1 | 1 | Input 3 |
| 1 | 0 | 0 | Input 4 |
| 1 | 0 | 1 | Input 5 |
| 1 | 1 | 0 | Input 6 |
| 1 | 1 | 1 | Input 7 |

Pengoperasian ADC 0809 dijelaskan dengan diagram pewaktuan atau “*timing diagram*” yang dapat dilihat dalam lampiran.

ADC 0809 memerlukan acuan Vref untuk referensi tegangan masukan. Guna memudahkan rangkaian, -Vref disambung ke *ground* dan +Vref disambung ke Vcc (catu +5 volt). Selanjutnya untuk mempermudah perhitungan, VCC dianggap tepat 5 volt atau 5000 mV, perhatikan gambar 2.5 untuk rangkain umum aplikasi ADC 0809.



**Gambar 2.5** **Rangkaian umum untuk aplikasi**

ADC 0809 bersifat rasiometrik, artinya ‘linier’ antara masukan, acuan dan keluaran. Masukan 5000 mV akan menghasilkan keluaran (angka) 255. masukan 0 mV akan menghasilkan keluaran 0. Jika +Vref misalnya diubah menjadi 4000 mV, maka kini masukan 4000 mV akan menghasilkan keluaran 255, demikian seterusnya.

Karena +Vref disambung ke Vcc (5000 mV), maka satu angka (LSB) setara dengan masukan 5000 / 256 atau 19,53 mV yang selanjutnya dianggap 20 mV, nilai inilah merupakan ketelitian ADC yang akan dibuat.

EOC bertugas memberitahu bahwa ADC telah selesai mengkonversi (yang juga berarti selesai mengukur) sehingga boleh dibaca keluarannya. Biasanya EOC dipergunakan untuk membangkitkan sela perangkat keras agar CPU mengetahui bahwa hasil ukur siap dibaca. Spesifikasi ADC 0809 menunjukkan bahwa lama konversi maksimum adalah 100 mikrodetik. Jika program tidak membutuhkan waktu yang singkat, maka program dapat menunggu, katakanlah 1 milidetik, lalu program sudah dapat membaca hasil dari pengukuran sehingga program dan rangkaian tidak lagi memerlukan sinyal EOC.

Sinyal detak atau clock diperlukan oleh ADC 0809 untuk operasionalnya, ADC 0809 akan beroperasi dengan clock mulai dari 10 KHZ s/d 1200 KHZ. Untuk menghasilkan clock dengan frekuensi tersebut maka dapat menggunakan IC NE/SE 555.

* 1. **PEWAKTU (*TIMER*) NE/SE 555**

Rangkaian pewaktu monolit NE/SE 555 adalah pengatur yang mantap yang mampu membangkitkan tundaan waktu sesuai dengan yang diinginkan hanya dengan merubah nilai resistor dan capasitornya.

Sifat-sifat dari NE/SE 555 :

* Waktu mati (*off*) kurang dari 12 udet.
* Frekuensi operasi tertinggi besar dari 500 kHZ.
* Pewaktu (*timing*) dari mikrodetik hingga jam.
* Beroperasi dalam ragam takstabil dan monostabil.
* Arus keluaran tinggi.
* Daur aktif (*duty cycle*) dapat distel.
* Serba cocok dengan TTL.
* Kemantapan suhu 0,005 %.

Ragam yang digunakan untuk memberikan sinyal detak bagi ADC 0809 adalah ragam takstabil. Dapat dilihat bentuk rangkaian pewaktu ini ialah multivibrator (*Astable Multivibrator*) ragam takstabil adalah seperti berikut :



**Gambar 2.6 NE/SE 555 Takstabil**

Frekuensi detak yang dihasilkan oleh rangkaian multivibrator ini adalah :

T = 0,7 ( R1 + 2R2 ) C2

f = =

Pin nomor 3 merupakan output dari NE/SE 555 yang akan dikoneksikan dengan pin 10 dari adc 0809 untuk frekuensi detak tertentu.

* 1. **IC 74LS00**

IC ini merupakan aplikasi dari gerbang logika NAND dimana digunakan untuk memberikan nilai kepada OE (*output enable*) dengan mengkombinasikan nilai dari pin START/ALE dan pin EOC (*end of conversion*)

* 1. **INTERFACE PORT PARALEL DENGAN ADC 0809**

Untuk proses *interface* (antarmuka) antara port paralel dengan ADC 0809, yang harus ditentukan terutama adalah berapa input dan output yang akan dikendalikan oleh komputer. Untuk ADC 0809 yang akan menjadi input ke komputer adalah 8 port data dan EOC (*end of conversion*) sedangkan yang menjadi output dari komputer adalah Address dan Start / ALE (*address latch enable*).

**Tabel 2.3**  **Koneksi pin port paralel dengan ADC 0809**

|  |  |  |
| --- | --- | --- |
| Pin Port Paralel | Pin ADC 0809 | Keterangan |
| 1 | 25 | Port Kontrol 🡪 Address A |
| 2 | 17 | Port Data 0 🡪 Data 0 (LSB) |
| 3 | 14 | Port Data 1 🡪 Data 1 |
| 4 | 15 | Port Data 2 🡪 Data 2 |
| 5 | 8 | Port Data 3 🡪 Data 3 |
| 6 | 18 | Port Data 4 🡪 Data 4 |
| 7 | 19 | Port Data 5 🡪 Data 5 |
| 8 | 20 | Port Data 6 🡪 Data 6 |
| 9 | 21 | Port Data 7 🡪 Data 7 (MSB) |
| 13 | 7 | Port Status 🡪 End Of Conversion |
| 14 | 24 | Port Kontrol 🡪 Address B |
| 16 | 23 | Port Kontrol 🡪 Address C |
| 17 | 6 / 22 | Port Kontrol 🡪 Start / ALE |
| 25 | 13 | Ground |

Untuk lebih jelasnya dapat dilihat dalam gambar 2.7 yang mana digambarkan secara skematik koneksi antara pin port paralel dengan pin dari ADC 0809.

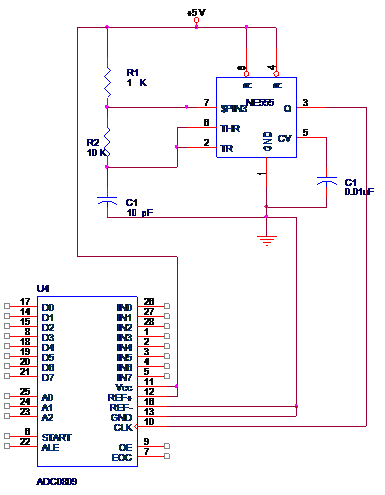


**Gambar 2.7 Interface ADC 0809 dengan port paralel**

Kombinasi dari pin keluaran port paralel untuk pin START/ALE dan port masukan ke port paralel untuk pin EOC akan menghasilkan masukan ke pin OE sehingga OE akan aktif jika EOC berlogika 1 atau high dan START/ALE berlogika 0 atau rendah.

* 1. **HUBUNGAN ADC 0809 DENGAN NE/SE 555**

Agar ADC 0809 dapat beroperasi dan mengirimkan serta mengambil data port paralel maka ADC harus mendapatkan sinyal clock yang berada pada pin nomor 10, clock yang dibutuhkan adalah sekita 500 KHZ, oleh karena itu rangkaian NE555 diseting agar menghasilkan clock yang mendekati 500 KHZ. Gambar 2.8 menunjukkan hubungan antara ADC 0809 dengan NE555.



**Gambar 2.8 Hubungan ADC 0809 dengan NE 555**

* 1. **DIAGRAM ALIR (*FLOW CHART*) PROGRAM UTAMA**

Agar mempermudah dalam pembuatan program pembacaan ADC 0809, terlebih dahulu dirancang diagram alir (*flowchart*), bentuk dari diagram alir program utama yang digunakan untuk proses pembacaan data dari ADC 0809 dapat dilihat dalam gambar 2.9 berikut :



**Gambar 2.9 Diagram alir program utama**