

PERKENALAN MIKROKONTROLER

1. Apakah Yang Disebut Dengan Mikrokontroler?

Suatu kontroler digunakan untuk mengontrol suatu proses atau aspek-aspek dari lingkungan. Satu contoh aplikasi dari mikrokontroler adalah untuk memonitor rumah kita. Ketika suhu naik kontroler membuka jendela dan sebaliknya.

Pada masanya, kontroler dibangun dari komponen-komponen logika secara keseluruhan, sehingga menjadikannya besar dan berat. Setelah itu barulah dipergunakan mikroprosesor sehingga keseluruhan kontroler masuk kedalam PCB yang cukup kecil. Hingga saat ini masih sering kita lihat kontroler yang dikendalikan oleh mikroprosesor biasa (Zilog Z80, Intel 8088, Motorola 6809, dsb).

Proses pengecilan komponen terus berlangsung, semua komponen yang diperlukan guna membangun suatu kontroler dapat dikemas dalam satu keping. Maka lahirlah komputer keping tunggal (one chip microcomputer) atau disebut juga mikrokontroler. Mikrokontrolere adalah suatu IC dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan untuk suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari:

1. CPU (Central Processing Unit)
2. RAM (Random Access Memory)
3. EEPROM/EPROM/PROM/ROM
4. I/O, Serial & Parallel
5. Timer
6. Interrupt Controller

Rata-rata mikrokontroler memiliki instruksi *manipulasi bit, akses ke I/O secara langsung dan mudah, dan proses interrupt yang cepat dan efisien*. Dengan kata lain mikrokontroler adalah " Solusi satu Chip" yang secara drastis mengurangi jumlah komponen dan biaya disain (harga relatif rendah).

2. Aplikasi Yang Dapat Dilakukan

Selain sebagai sistem monitor rumah seperti diatas, mikrokontroler sering dijumpai pada peralatan rumah tangga (microwave oven, TV, stereo set dll), komputer dan perlengkapannya, mobil dan lain sebagainya. Pada beberapa penggunaan bisa ditemukan lebih dari satu prosesor didalamnya.

Mikrokontroler biasanya digunakan untuk peralatan yang tidak terlalu membutuhkan kecepatan pemrosesan yang tinggi. Walaupun mungkin ada diantara kita yang membayangkan untuk mengontrol oven microwave dengan menggunakan sistem berbasis Unix, mengendalikan oven microwave dapat dengan mudah menggunakan mikrokontroler yang paling kecil. Dilain pihak jika kita ingin mengendalikan rudal guna mengejar anjing tetangga yang selalu menyalak ditengah malam, kita akan memerlukan prosesor dengan kecepatan yang lebih tinggi.

Sifat spesial dari mikrokontroler adalah *kecil dalam ukuran, hemat daya listrik serta fleksibilitasnya* menyebabkan mikrokontroler sangat cocok untuk dipakai sebagai pencatat/perekam data pada aplikasi yang tidak memerlukan kehadiran operator.

3. Jenis Yang Dapat Dipilih

Mikrokontroler tersedia dalam beberapa pilihan, tergantung dari keperluan dan kemampuan yang diinginkan. Kita dapat memilih mikrokontroler 4, 8, 16 atau 32 bit. Disamping itu terdapat pula mikrokontroler dengan kemampuan komunikasi serial, penanganan keyboard, pemroses sinyal, pemroses video dll.

4. Pasar Bagi Mikrokontroler

WorldWide Microcontroller Shipments (in millions of dollars)

	90	91	92	93	94	95	96	97	98	99	00
4 bit	1,393	1,597	1,596	1,698	1,761	1,826	1,849	1,881	1,856	1,816	1,757
8 bit	2,077	2,615	2,862	3,703	4,689	5,634	6,553	7,529	8,423	9,219	9,715
16 bit	192	303	340	484	810	1,170	1,628	2,191	2,969	3,678	4,405

WorldWide Microcontroller Shipments (in Millions)

	90	91	92	93	94	95	96	97	98	99	00
4 bit	778	906	979	1,036	1,063	1110	1110	1096	1,064	1,025	970
8 bit	588	753	843	1,073	1,449	1,803	2,123	2,374	2,556	2,681	2,700
16 bit	22	38	45	59	106	157	227	313	419	501	585

Source: WSTS & ICE - 1994

Jika kita bertanya apa perlunya kita mempelajari mikrokontroler, tabel diatas akan sedikit banyak memberikan gambaran tentang bisnis mikrokontroler yang akan menghasilkan banyak tumpukan rupiah dimeja kita. Suatu survey di Amerika menyatakan bahwa rata-rata terdapat 35 buah mikrokontroler yang digunakan pada satu rumah di Amerika, dan diperkirakan akan menjadi 240 pada tahun 2000.

5. Pertimbangan Pemilihan Mikrokontroler

Terdapat beberapa hal yang perlu dipertimbangkan dalam menentukan jenis mana yang akan dipergunakan dalam disain kita yaitu seperti berikut:

- ❖ Ketersediaan dan harga dari suatu development tools (Programmer, Emulator dan Simulator)
- ❖ Ketersediaan dokumentasi (Ref. Manual, Application notes, dan buku lainnya).
- ❖ Ketersediaan tempat bertanya.
- ❖ Ketersediaan komponen OTP, Mask, dan Programmable.

6. Produsen Mikrokontroler

Dibawah ini adalah daftar produsen besar dari mikrokontroler dan unit yang terjual (dalam ribuan).

Company (Units x 1000)

Motorola (358,894)
Mitsubishi(71,674)
NEC (70,180)
Hitachi (67,873)
Philips (56,680)
Intel (46,876)
SGS-Thomson (37,350)
Microchip (35,477)
Matsushitta (34,200)
Toshiba (32,205)
National Semiconductor (31,634)
Zilog (31,000)
Texas Instruments (29,725)
Siemens (20,874)
Sharp(17,505)

SOURCE: DataQuest June 1994.

7. Hal-Hal Mengenai Mikrokontroler

❖ **Tehnik fabrikasi**

CMOS - Complementary Metal Oxide Semiconductor

Ini adalah tehnik yang biasa dilakukan untuk memproduksi hampir semua mikrokontroler terbaru. Mikrokontroler CMOS memerlukan daya yang lebih rendah dibanding mikrokontroler yang dibuat dengan tehnik sebelumnya, sehingga memungkinkan untuk dioperasikan menggunakan batere. Chip CMOS juga memungkinkan dioperasikan pada fully atau mendekati fully static, yang berarti bahwa clock dapat diperlambat bahkan diberhentikan sehingga chip berada dalam kondisi (mode) sleep. CMOS juga lebih tahan terhadap noise dibandingkan cara fabrikasi sebelumnya.

❖ **Arsitektur**

Von-Neuman Architecture

Mikrokontroler yang di disain berdasarkan arsitektur ini memiliki sebuah data bus yang dipergunakan untuk "fetch" instruksi dan data. Program (instruksi) dan data disimpan pada memori utama secara bersama-sama. Ketika kontroler mengamati suatu alamat di memori utama, hal pertama yang dilakukan adalah mengambil instruksi untuk dilaksanakan dan kemudian mengambil data pendukung dari instruksi tsb. Cara ini memperlambat operasi mikrokontroler.

Harvard Architecture

Arsitektur ini memiliki bus data dan instruksi yang terpisah, sehingga memungkinkan eksekusi dilakukan secara bersamaan. Secara teoritis hal ini memungkinkan eksekusi yang lebih cepat tetapi dilain pihak memerlukan disain yang lebih kompleks.

❖ **Instruksi**

CISC

Saat ini hampir semua mikrokontroler adalah mikrokontroler CISC (Complete Instruction Set Computer). Biasanya memiliki lebih dari 80 instruksi. Keunggulan dari CISC ini adalah adanya instruksi yang bekerja seperti sebuah makro, sehingga memungkinkan programmer untuk menggunakan sebuah instruksi menggantikan beberapa instruksi sederhana lainnya.

RISC

Saat ini kecenderungan industri untuk menggunakan disain mikroprosesor RISC (Reduced Instruction Set Computer). Dengan menggunakan jumlah instruksi yang lebih sedikit, memungkinkan lahan pada chip (silicon real-estate) digunakan untuk meningkatkan kemampuan chip. Keuntungan dari RISC adalah kesederhanaan disain, chip yang lebih kecil, jumlah pin sedikit dan sangat sedikit mengkonsumsi daya.

Dibawah ini adalah fature yang biasa dimiliki oleh RISC Processor:

Harvard Architecture, memungkinkan akses yang program dan data yang bersamaan .
Instruction Pipelining meningkatkan kecepatan eksekusi.

Orthogonal instruction set untuk kemudahan dalam programming, memungkinkan tiap instruksi untuk dioperasikan pada register atau digunakan pada beberapa mode pengalamatan, instruksi-instruksi tidak mempunyai kombinasi tertentu dan juga tanpa perkecualian.

8. Pilihan Memori

EEPROM - Electrically Erasable Programmable Read Only Memory

Beberapa mikrokontroler memiliki EEPROM yang terintegrasi pada chipnya. EEPROM ini digunakan untuk menyimpan sejumlah kecil parameter yang dapat berubah dari waktu ke waktu. Jenis memori ini bekerja relatif pelan, dan kemampuan untuk dihapus/tulis nya juga terbatas.

FLASH (EPROM)

FLASH memberikan pemecahan yang lebih baik dari EEPROM ketika dibutuhkan sejumlah besar memori non-volatile untuk program. FLASH ini bekerja lebih cepat dan dapat dihapus/tulis lebih sering dibanding EEPROM.

Battery backed-up static RAM

Memori ini sangat berguna ketika dibutuhkan memori yang besar untuk menyimpan data dan program. Keunggulan utama dari RAM statis adalah sangat cepat dibanding memori non-volatile, dan juga tidak terdapat keterbatasan kemampuan hapus/tulis sehingga sangat cocok untuk aplikasi untuk menyimpan dan manipulasi data secara lokal.

Field programming/reprogramming

Dengan menggunakan memori non-volatile untuk menyimpan program akan memungkinkan mikrokontroler tersebut untuk diprogram ditempat, tanpa melepaskan dari sistem yang dikontrolnya. Dengan kata lain mikrokontroler tersebut dapat diprogram setelah dirakit diPCBnya.

OTP - One Time Programmable

Mikrokontroler OTP adalah mikrokontroler yang hanya dapat diprogram satu kali saja dan tidak dapat dihapus atau dimodifikasi. Biasanya digunakan untuk produksi dengan jumlah terbatas. OTP menggunakan EPROM standard tetapi tidak memiliki jendela untuk menghapus programnya.

Software protection

Dengan "encryption" atau proteksi fuse, software yang telah diprogramkan akan terlindungi dari pembajakan, modifikasi atau rekayasa ulang. Kemampuan ini hanya dimiliki oleh komponen OTP atau komponen yang dapat diprogram ulang. Pada komponen jenis Mask ROM tidak diperlukan proteksi, hal ini dikarenakan untuk membajak isi programnya seseorang harus membacanya (visual) dari chip nya dengan menggunakan mikroskop elektron.

Walaupun demikian pabrik mikrokontroler masih dapat membaca isi program guna memastikan bahwa mikrokontroler diprogram dengan tepat, atau biasa disebut "test mode". TEST MODE MEMUNGKINKAN KITA MEMBACA KESELURUHAN ISI ROM , tetapi hal ini tidak perlu dibesar-besarkan karena test mode ini bersifat SANGAT-SANGAT- SANGAT DIRAHASIKAN dan hanya diketahui oleh pabrikan yang memproduksi mikrokontroler tersebut. Test mode hanya dapat dilakukan pada komponen Mask ROM.

9. Input/Output

UART (Universal Asynchronous Receiver Transmitter) adalah adapter serial port adapter untuk komunikasi serial asinkron.

USART (Universal Synchronous/Asynchronous Receiver Transmitter) merupakan adapter serial port untuk komunikasi serial sinkron dan asinkron. Komunikasi serial sinkron tidak memerlukan start/stop bit dan dapat beroperasi pada clock yang lebih tinggi dibanding asinkron.

SPI (serial peripheral interface) merupakan port komunikasi serial sinkron.

SCI (serial communications interface) merupakan enhanced UART (asynchronous serial port)

I2C bus (Inter-Integrated Circuit bus) merupakan antarmuka serial 2 kawat yang dikembangkan oleh Philips. Dikembangkan untuk aplikasi 8 bit dan banyak digunakan pada consumer electronics, automotive dan industri. I2C bus ini berfungsi sebagai antarmuka jaringan multi-master, multi-slave dengan deteksi tabrakan data. Jaringan dapat dipasangkan hingga 128 titik dalam jarak 10 meter. Setiap titik dalam jaringan dapat mengirim dan menerima data. Setiap titik dalam jaringan harus memiliki alamat yang unik.

Analog to Digital Conversion (A/D). Fungsi ADC adalah merubah besaran analog (biasanya tegangan) ke bilangan digital. Mikrokontroler dengan fasilitas ini dapat digunakan untuk aplikasi-aplikasi yang memerlukan informasi analog (misalnya voltmeter, pengukur suhu dll).

Terdapat beberapa tipe dari ADC sbb:

- Successive Approximation A/D converters.
- Single Slope A/D converters.
- Delta-Sigma A/Ds converters.
- Flash A/D.

D/A (Digital to Analog) Converters. Kebalikan dari ADC seperti diatas.

Comparator. Mikrokontroler tertentu memiliki sebuah atau lebih komparator. Komparator ini bekerja seperti IC komparator biasa tetapi sinyal input/outputnya terpasang pada bus mikrokontroler.

10. Interupsi

Interrupt merupakan metode yang efisien bagi mikrokontroler untuk memproses periperalnya, mikrokontroler hanya bekerja memproses periperall tsb hanya pada saat terdapat data diperiperall tsb. Pada saat terjadi interrupt, mikrokontroler menunda operasi yang sedang dilakukan kemudian mengidentifikasi interupsi yang datang dan menjalankan rutin pelayanan interupsi.

Rata-rata mikrokontroler memiliki setidaknya-tidaknya sebuah interupsi eksternal, interupsi yang dimiliki bisa dipicu oleh "edge" atau "level". Edge triggered interrupt bekerja tidak tergantung pada waktu terjadinya interupsi, tetapi interupsi bisa terjadi karena glitch. Sedangkan Level triggered interrupt harus tetap pada logika high atau low sepanjang waktu tertentu agar dapat terjadi interupsi, interupsi ini tahan terhadap glitch.

Maskable Interrupts

Dengan maskable interrupt kita dapat bebas memilih untuk menggunakan satu atau lebih interupsi. Keuntungan maskable interrupt ini adalah kita dapat mematikan interupsi pada saat mikrokontroler sedang melakukan proses yang kritis sehingga interupsi yang datang akan diabaikan.

Vectored Interrupts

Pada saat terjadi interupsi, interrupt handler secara otomatis akan memindahkan program pada alamat tertentu yang telah ditentukan sesuai dengan jenis interupsi yang terjadi.

11. Mikrokontroler Populer

Dibawah ini dijelaskan beberapa mikrokontroler yang cukup populer. Untuk menggunakan salah satu mikrokontroler ini pilihan yang paling tepat adalah mikrokontroler yang memiliki dokumentasi yang baik serta development tools dengan harga yang terjangkau. Untuk pemula atau hobyst, Intel 8051, Motorola 68hc11 atau Microchip PIC adalah pilihan yang cukup baik.

❖ 8051 (Intel dan lainnya)

Arsitektur Harvard modified dengan alamat terpisah untuk memori program dan data. Memori untuk program bisa dialamati hingga 64 K. Memori bawah (4K, 8K atau 16K tergantung tipe) bisa terletak di chipnya. Mikrokontroler ini memiliki 128 byte memori internal ditambah beberapa register (SFR), juga bisa mengalami hingga 64K memori eksternal untuk data.

Cukup banyak software baik software komersil maupun gratis untuk mikrokontroler 8051 ini. Mikrokontroler ini memiliki banyak varian sehingga mampu memenuhi keperluan yang berbeda. Diproduksi tidak hanya oleh Intel tetapi beberapa pabrikan lainnya juga ikut memproduksi jenis mikrokontroler ini.

❖ 6805 (Motorola)

Memiliki arsitektur Von Neuman dimana instruksi, data, I/O, dan timer terdapat pada satu daerah memori. Stack pointer yang dimiliki adalah 5 bit sehingga kedalaman stack terbatas hingga 32 byte. Beberapa mikrokontroler dari keluarga ini memiliki ADC, PLL, Frq. Synthesizer, serial I/O dan software security.

❖ PIC (MicroChip)

Mikrokontroler PIC merupakan mikrokontroler RISC yang pertama. Pada umumnya RISC mengakibatkan kesederhanaan rancangan dan memungkinkan untuk menambah kemampuannya dengan biaya yang rendah. Walaupun hanya memiliki sedikit instruksi (33 instruksi untuk 16C5x), keluarga PIC memiliki banyak keunggulan yang sudah merupakan bagian dari chip. Dengan bus instruksi dan bus data yang terpisah (arsitektur Harvard), PIC memungkinkan akses data dan program secara bersamaan sehingga menaikkan kinerja pemrosesannya. Keuntungan dari kesederhanaan rancangan ini adalah chip yang sangat kecil, sedikit pin dan pemakaian daya yang sangat kecil.

Popularitas mikrokontroler PIC ini meningkat sangat cepat. Dengan harga yang murah, ukuran kecil dan hemat pemakaian daya, pada saat ini mikrokontroler ini digunakan juga pada pemakaian lain seperti sebagai rangkaian logika. Terdapat tiga keluarga PIC pada saat ini yaitu PIC16C5x, PIC16Cxx dan PIC17Cxx.

❖ Z8 (Zilog)

Z8 merupakan turunan dari Zilog Z80. Memiliki arsitektur unik merupakan arsitektur gabungan dengan tiga daerah memori yaitu: program memori, data memori dan CPU register file. Mikrokontroler ini memiliki UART, timer, DMA, I/O hingga 40 buah pada chipnya. Versi lainnya memiliki sync/async serial channel.

Keseluruhan mikrokontroler ini memiliki Stack RAM yang dapat dikonfigurasi dan sistem interupsi, dua timer programmable dengan interrupt, proteksi ROM, dua analog komparator

12. Software

Software untuk menggunakan mikrokontroler dapat didapatkan secara gratis dengan mencari di WWW, tetapi software gratis biasanya tidak dilengkapi dengan dokumentasinya. Software ini diantaranya adalah software untuk simulasi dan software untuk pemrograman. Jika ingin lebih serius mendalami mikrokontroler mungkin lebih baik untuk membeli software lengkap dengan dokumentasinya.

Jika PIC merupakan mikrokontroler yang dipilih, kita dapat mendownload software yang diperlukan secara gratis di web microchip yaitu MPSIM (simulator), MPASM (assembler).

13. Bahasa Pemrograman Bagi Mikrokontroler

BAHASA MESIN DAN ASSEMBLER

Bahasa mesin adalah satu-satunya bahasa yang dimengerti oleh mikrokontroler. Bahasa ini tidak mudah untuk dimengerti oleh manusia. Sedangkan bahasa assembly adalah suatu bentuk bahasa mesin yang bisa dimengerti oleh manusia. Setiap pernyataan dari bahasa assembly menggambarkan satu pernyataan bahasa mesin. Sebagai contoh instruksi JMP (asal kata JUMP) akan lebih mudah dimengerti dibandingkan instruksi B3H.

Pemrograman dengan menggunakan bahasa assembly/mesin menghasilkan program yang kecil dan cepat. Hal ini dikarenakan kita sepenuhnya mengontrol kerja dari program, tetapi tentu saja jika kita membuat program yang bertele-tele dan berbelit akan menyebabkan program berjalan lambat.

Untuk orang yang pertama kali mempelajari mikrokontroler, akan lebih baik jika mempelajari assembler terlebih dahulu sebelum mempelajari bahasa pemrograman lainnya (mis: C). Dengan membuat program dengan assembler akan membimbing kita memahami arsitektur dari mikrokontroler tsb.

KOMPILER

Compiler adalah penerjemah untuk bahasa pemrograman tingkat tinggi. Bekerja dengan cara menterjemahkan (mis pada PC) langsung ke bahasa mesin yang dimengerti oleh mikrokontroler.

Salah satu compiler yang banyak dipergunakan saat ini adalah "C". "C" digunakan pada mikrokontroler kecil hingga supercomputer. Walaupun program dengan C sedikit sulit untuk dipahami (diakibatkan oleh gaya penulisan program yang berbeda untuk tiap programmer), C merupakan alat yang sangat flexible dan sangat membantu pengembangan program. Bahasa ini adalah bahasa tingkat tinggi tetapi masih memungkinkan kita akses langsung ke mesin. Saat ini terdapat beberapa compiler C yang cukup murah dan bagus untuk pemrograman mikrokontroler terkenal. Kode (bahasa mesin) yang dihasilkan oleh compiler ini cukup efisien (cepat dan kompak).

14. Alat Bantu Pengembangan

Memiliki software pemrograman belum mencukupi untuk mengembangkan program bagi suatu mikrokontroler. Diperlukan pula software untuk mencari kesalahan dalam pemrograman sbb:

SIMULATOR

Fungsi simulator adalah mensimulasikan atau menirukan kerja mikrokontroler pada PC. Langkah-langkah yang dikerjakan serta apa yang terjadi ketika program dijalankan dapat diamati dilayar PC. Disamping itu juga isi dari register atau variabel dapat diisi atau diubah ketika program dijalankan. Simulator tidak dapat mensimulasi kehadiran interupsi secara baik, dan biasanya program yang dijalankan jauh lebih lambat dibandingkan pada keadaan sebenarnya.

DEBUGGER RESIDEN

Debugger residen menjalankan program di mikrokontroler itu sendiri, dan pada saat bersamaan menampilkan hasilnya pada komputer induknya (PC). Alat bantu ini memiliki beberapa keunggulan seperti pada simulator dengan kelebihan lain yaitu kita dapat melihat bagaimana program tersebut bekerja pada target yang sebenarnya. Namun disisi lain, alat bantu ini memakai sebagian sumber daya yang dimiliki oleh mikrokontroler seperti port komunikasi (untuk komunikasi dgn PC), interupsi untuk menjalankan program perlangkah (single step) dan sejumlah memori untuk menyimpan program dari debugger (bagian residen yang ditempatkan di target).

EMULATOR

Emulator adalah peralatan yang bekerja dengan berpura-pura sebagai mikrokontroler dan pada saat bersamaan dia mengambil informasi untuk ditampilkan. Emulator memberikan kontrol penuh pada target. Emulator ini bisa berupa perangkat dengan display tersendiri atau merupakan pengantar muka PC.

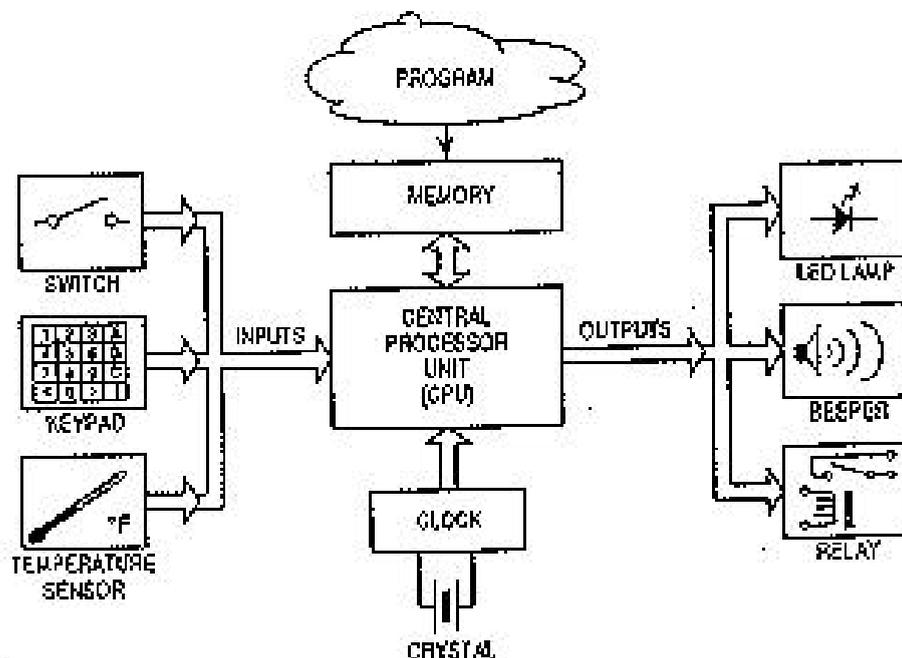
Jika cukup banyak dana yang dimiliki, emulator ini adalah alat yang benar-benar diperlukan dalam mengembangkan suatu sistem.

Konsep Mikrokontroler

A. Pendahuluan

Mikrokontroler adalah salah satu dari bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan output spesifik berdasarkan inputan yang diterima dan program yang dikerjakan.

Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer. Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh programmer.



B. Sistem Input Komputer

Piranti input menyediakan informasi kepada sistem komputer dari dunia luar. Dalam sistem komputer pribadi, piranti input yang paling umum adalah keyboard. Komputer mainframe menggunakan keyboard dan pembaca kartu berlubang sebagai piranti inputnya. Sistem dengan mikrokontroler umumnya menggunakan piranti input yang jauh lebih kecil seperti saklar atau keypad kecil.

Hampir semua input mikrokontroler hanya dapat memproses sinyal input digital dengan tegangan yang sama dengan tegangan logika dari sumber. Level nol disebut dengan VSS dan tegangan positif sumber (VDD) umumnya adalah 5 volt. Padahal dalam dunia nyata terdapat banyak sinyal analog atau sinyal dengan tegangan level yang bervariasi. Karena itu ada piranti input yang mengkonversikan sinyal analog menjadi sinyal digital sehingga komputer bisa mengerti dan menggunakannya. Ada beberapa mikrokontroler yang dilengkapi dengan piranti konversi ini, yang disebut dengan ADC, dalam satu rangkaian terpadu.

C. Sistem Output Komputer

Piranti output digunakan untuk berkomunikasi informasi maupun aksi dari sistem komputer dengan dunia luar. Dalam sistem komputer pribadi (PC), piranti output yang umum adalah monitor CRT. Sedangkan sistem mikrokontroler mempunyai output yang jauh lebih sederhana seperti lampu indikator atau beeper. Frasa kontroler dari kata mikrokontroler memberikan penegasan bahwa alat ini mengontrol sesuatu.

Mikrokontroler atau komputer mengolah sinyal secara digital, sehingga untuk dapat memberikan output analog diperlukan proses konversi dari sinyal digital menjadi analog. Piranti yang dapat melakukan konversi ini disebut dengan DAC (Digital to Analog Converter).

D. CPU (Central Processing Unit)

CPU adalah otak dari sistem komputer. Pekerjaan utama dari CPU adalah mengerjakan program yang terdiri atas instruksi-instruksi yang diprogram oleh programmer. Suatu program komputer akan menginstruksikan CPU untuk membaca informasi dari piranti input, membaca informasi dari dan menulis informasi ke memori, dan untuk menulis informasi ke output.

Dalam mikrokontroler umumnya hanya ada satu program yang bekerja dalam suatu aplikasi. CPU M68HC05 mengenali hanya 60 instruksi yang berbeda. Karena itu sistem komputer ini sangat cocok dijadikan model untuk mempelajari dasar dari operasi komputer karena dimungkinkan untuk menelaah setiap operasi yang dikerjakan.

E. Clock dan Memori Komputer

Sistem komputer menggunakan osilator clock untuk memicu CPU mengerjakan satu instruksi ke instruksi berikutnya dalam alur yang berurutan. Setiap langkah kecil dari operasi mikrokontroler memakan waktu satu atau beberapa clock untuk melakukannya.

Ada beberapa macam tipe dari memori komputer yang digunakan untuk beberapa tujuan yang berbeda dalam sistem komputer. Tipe dasar yang sering ditemui dalam mikrokontroler adalah ROM (Read Only Memory) dan RAM (Random Access Memory). ROM digunakan sebagai media penyimpan program dan data permanen yang tidak boleh berubah meskipun tidak ada tegangan yang diberikan pada mikrokontroler. RAM digunakan sebagai tempat penyimpanan data sementara dan hasil kalkulasi selama proses operasi. Beberapa mikrokontroler mengikutsertakan tipe lain dari memori seperti EPROM (Erasable Programmable Read Only Memory) dan EEPROM (Electrically Erasable Programmable Read Only Memory).

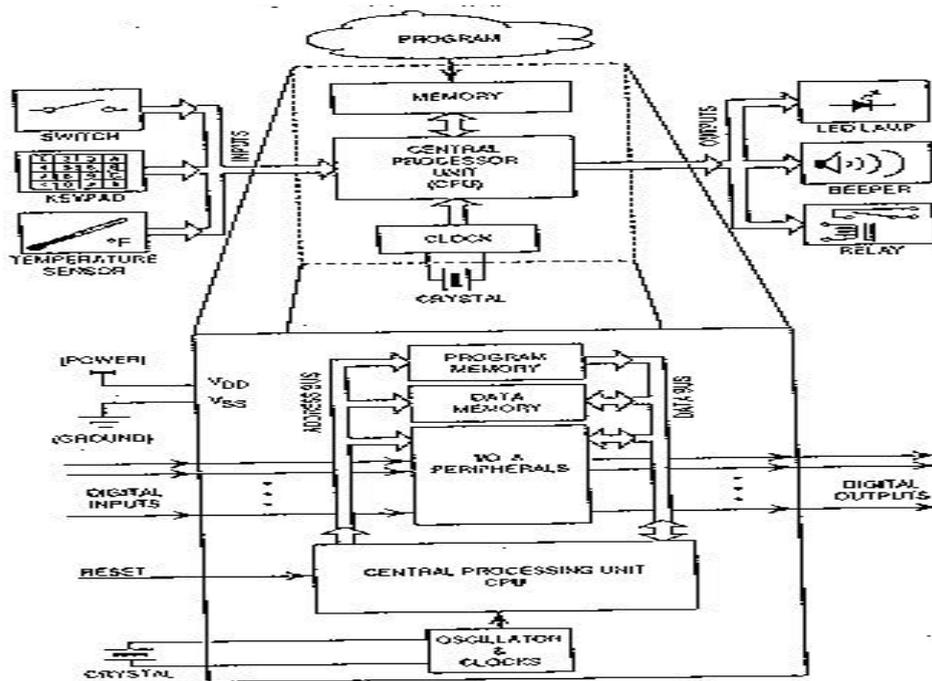
F. Program Komputer

Program digambarkan sebagai awan karena sebenarnya program adalah hasil imajinasi seorang programmer. Komponen utama dari program adalah instruksi-instruksi dari instruksi set CPU. Program disimpan dalam memori dalam sistem komputer di mana mereka dapat secara berurutan dikerjakan oleh CPU.

G. Sistem Mikrokontroler

Setelah dipaparkan bagian-bagian dari suatu sistem komputer, sekarang akan dibahas mengenai mikrokontroler. Digambarkan sistem komputer dengan bagian yang dikelilingi oleh garis putus-putus. Bagian inilah yang menyusun mikrokontroler. Bagian yang dilingkupi kotak bagian bawah adalah gambar lebih detail dari susunan bagian yang dilingkupi garis putus-putus. Kristal tidak termasuk dalam sistem mikrokontroler tetapi diperlukan dalam sirkuit osilator clock.

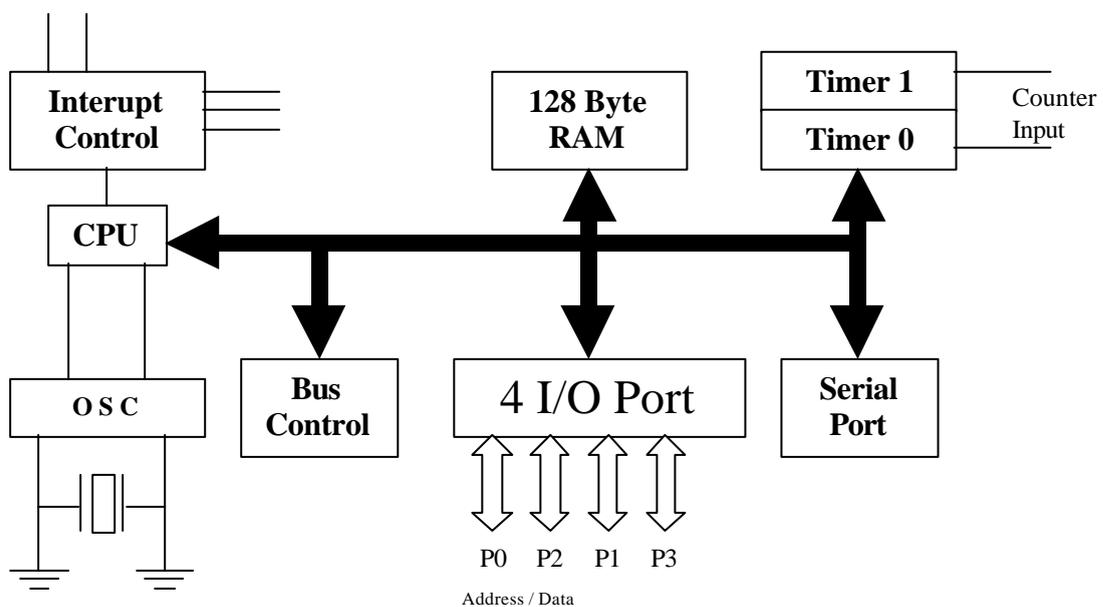
Suatu mikrokontroler dapat didefinisikan sebagai sistem komputer yang lengkap termasuk sebuah CPU, memori, osilator clock, dan I/O dalam satu rangkaian terpadu. Jika sebagian elemen dihilangkan, yaitu I/O dan memori, maka chip ini akan disebut sebagai mikroprosesor.



H. Mikrokontroler Keluarga 8051

Mikrokontroler 8051 merupakan keluarga mikrokontroler MCS-51. Yang termasuk dalam keluarga MCS-51 adalah mikrokontroler 8031 (versi 8051 tanpa EPROM), 8751, dan 8052. Keluarga MCS-51 memiliki tipe CPU, RAM, counter/ timer, port paralel, dan port serial yang sama. Mikrokontroler 8051 diperkenalkan pertama kali oleh Intel corp. pada akhir 1970. Mikrokontroler 8051 merupakan controller 8-bit yang mampu mengakses 64 Kbyte memory dan 64 Kbyte data memory (eksternal).

I. Blok diagram 8051:



Fasilitas yang terdapat dalam chip mikrokontroler 8051 diantaranya adalah:

- ❖ 2 timer/counter → XTAL 1 & XTAL2
- ❖ 4 port paralel input/output → Port 0, Port 1, Port 2 & Port 3
- ❖ 5 sumber interrupt control logic → INT0, INT1, T0, T1 & Serial Port

J. Register Utama

Keluarga MCS-51 memiliki banyak register yang meliputi akumulator, register R, register B, DPTR (Data Pointer), PC (Program Counter), dan SP (Stack Pointer).

Akumulator

Akumulator, sebagaimana namanya, digunakan sebagai register umum untuk mengakumulasikan hasil dari instruksi-instruksi. Akumulator dapat menampung 8 bit (1 byte) data dan merupakan register yang paling sering dipakai. Hampir lebih dari separuh dari instruksi keluarga 8051 menggunakan akumulator. Sebagai contoh, jika ada operasi penjumlahan 10 dengan 20, yang menghasilkan 30, akan disimpan dalam akumulator.

Register R dan B

Register R adalah delapan set register yang dinamakan R0, R1, R2, R3, R4, R5, R6, dan R7. Register-register ini digunakan sebagai register pembantu penyimpanan data dalam banyak operasi. Tanpa register-register ini, akumulator tidak dapat melakukan operasi antara dua operan.

Register B sama dengan akumulator dalam hal dia menyimpan sebuah harga 8 bit (1 byte). Register B ini hanya digunakan dalam dua instruksi keluarga 8031, yaitu MUL AB dan DIV AB. Karenanya, jika diinginkan untuk mengalikan atau membagi akumulator A dengan suatu harga, maka simpanlah harga tersebut dalam register B dan kemudian jalankan instruksinya.

DPTR, PC, Dan SP

DPTR (Data Pointer) adalah satu-satunya register yang dapat diakses 16 bit (2 byte) di dalam keluarga 8031. Sebagaimana namanya, DPTR digunakan untuk menunjuk pada lokasi suatu data. DPTR digunakan oleh beberapa perintah yang mengizinkan keluarga 8031 untuk mengakses memori eksternal. Jika 89C51 mengakses eksternal memori, dia akan mengakses memori eksternal dengan alamat yang ditunjukkan oleh DPTR..

PC (Program Counter) adalah alamat 2 byte yang memberitahu 89C51 di mana instruksi selanjutnya akan dilaksanakan. Saat 89C51 inisialisasi, PC selalu berisi 0000h dan bertambah satu setiap satu instruksi terkerjakan. Penting untuk diingat bahwa PC selalu bertambah satu. Demikian pula PC tidak dapat dimodifikasi harganya secara langsung dengan menggunakan perintah MOV PC,2430h. Namun perintah LJMP 2340h akan mengisi PC dengan 2340h.

SP (Stack Pointer) mampu menyimpan 8 bit (1 byte) data. SP digunakan untuk menunjukkan di mana harga berikutnya yang akan diambil dari stack. Jika suatu harga dimasukkan dalam stack, 89C51 pertama-tama akan menambah harga SP dan kemudian menyimpan harga tersebut pada alamat memori yang bersesuaian. Demikian pula jika suatu harga diambil dari stack, maka 89C51 akan mengambil harga dari stack dan kemudian mengurangi harga SP.

K. Mode Pengalamatan

Mode pengalamatan merujuk pada bagaimana pemrogram mengalami suatu lokasi memori. Setiap mode pengalamatan memberikan fleksibilitas khusus yang sangat penting. Mode pengalamatan ini meliputi immediate addressing, direct addressing, dan indirect addressing.

1. Immediate Addressing

Mode pengalamatan immediate addressing sangat umum dipakai karena harga yang akan disimpan dalam memori langsung mengikuti kode operasi dalam memori. Dengan kata lain, tidak diperlukan pengambilan harga dari alamat lain untuk disimpan. Contohnya: `MOV A,#20h`. Dalam instruksi tersebut, akumulator akan diisi dengan harga yang langsung mengikutinya, dalam hal ini 20h. Mode ini sangatlah cepat karena harga yang dipakai langsung tersedia.

2. Direct Addressing

Dalam mode pengalamatan direct addressing, harga yang akan dipakai diambil langsung dalam alamat memori lain. Contohnya: `MOV A,30h`. Dalam instruksi ini akan dibaca data dari RAM internal dengan alamat 30h dan kemudian disimpan dalam akumulator. Mode pengalamatan ini cukup cepat, meskipun harga yang didapat tidak langsung seperti immediate, namun cukup cepat karena disimpan dalam RAM internal. Demikian pula akan lebih mudah menggunakan mode ini daripada mode immediate karena harga yang didapat bisa dari lokasi memori yang mungkin variabel.

3. Indirect Addressing

Mode pengalamatan indirect addressing sangat berguna karena dapat memberikan fleksibilitas tinggi dalam mengalami suatu harga. Mode ini pula satu-satunya cara untuk mengakses 128 byte lebih dari RAM internal pada keluarga 8052. Contoh: `MOV A,@R0`. Dalam instruksi tersebut, 89C51 akan mengambil harga yang berada pada alamat memori yang ditunjukkan oleh isi dari R0 dan kemudian mengisikannya ke akumulator.

Mode pengalamatan indirect addressing selalu merujuk pada RAM internal dan tidak pernah merujuk pada SFR. Karena itu, menggunakan mode ini untuk mengalami alamat lebih dari 7Fh hanya digunakan untuk keluarga 8052 yang memiliki 256 byte spasi RAM internal.

Mode pengalamatan memori eksternal menggunakan mode ini dan terdiri atas dua bagian.

- Bagian pertama digunakan untuk mengakses memori eksternal, di mana alamatnya terdapat di dalam DPTR 16 bit. Contohnya: `MOVX A,@DPTR` atau `MOVX @DPTR,A`.
- Bagian kedua mengakses alamat memori eksternal secara 8 bit (1 byte), di mana alamat dari harga yang akan diambil terdapat di dalam register R. Contohnya: `MOVX @R0,A`. Dalam instruksi tersebut, alamat yang terdapat di dalam register R0 dibaca dahulu dan kemudian harga akumulator ditulis dengan harga yang terdapat pada alamat memori eksternal yang didapat. Karena register R0 hanya dapat menampung alamat dari 00h hingga FFh, maka penggunaan mode pengalamatan eksternal 8 bit ini hanya terbatas pada 256 byte dari memori eksternal.

L. Timer

Mikrokontroler 89C51 hadir dengan dua timer, keduanya bisa dikontrol, diset, dibaca, dan dikonfigurasi sendiri-sendiri. Timer 89C51 memiliki tiga fungsi umum, yaitu:

- 1) Menghitung waktu antara dua kejadian (event)
- 2) Menghitung jumlah kejadian itu sendiri
- 3) Membangkitkan baud rate untuk port serial.

Sebuah timer bekerja dengan mencacah. Tidak tergantung pada fungsi sebagai timer, counter, atau generator baud rate, sebuah timer akan selalu ditambah satu oleh mikrokontroler.

Menggunakan Timer Untuk Mengukur Waktu

Fungsi timer yang utama adalah untuk mengukur waktu. Saat sebuah timer digunakan untuk mengukur waktu, dia akan bertambah satu setiap satu siklus mesin. Setiap siklus mesin membutuhkan 12 pulsa kristal. Maka, apabila sebuah 89C51 dengan kristal 11,059 MHz, maka timer setiap detiknya akan berharga:

$$11.059.000/12=921.583$$

Dengan kata lain, terdapat 921.583 kali pencacahan dalam setiap detiknya. Tidak seperti instruksi- instruksi yang bisa memakan satu hingga empat siklus mesin, sebuah timer selalu konsisten bertambah satu setiap satu kali siklus mesin. Sehingga, jika diinginkan sebuah timer yang berharga 50.000, berarti memakan waktu sebesar:

$$50.000/921.583=0,0542$$

Dengan kata lain, diperlukan waktu 0,052 detik untuk mendapatkan timer yang telah berharga 50.000. Melalui cara serupa, apabila diinginkan mendapatkan pewaktu 0,05 detik, maka dibutuhkan timer yang mencacah hingga:

$$0,05 \times 921.583=46.079,15$$

Ini berarti kita perlu memonitor cacahan dari timer hingga mencapai harga 46.079. Walaupun tidak benar-benar presisi karena menghilangkan hitungan 0,15; namun cukup mendekati dan dapat ditoleransi.

Timer SFR

Sebagaimana telah disinggung di atas, 89C51 memiliki dua buah timer yang setiap fungsinya identik. Timer pertama disebut dengan TIMER0 dan timer kedua disebut dengan TIMER1. Kedua timer saling berbagi dua macam SFR, yaitu TMOD dan TCON, yang mengontrol timer, dan masing-masing timer memiliki dua macam SFR yang spesifik yaitu TH0/TL0 untuk TIMER0 dan TH1/TL1 untuk TIMER1. Untuk lebih jelasnya lihat tabel di bawah ini.

Daftar SFR Untuk Timer

Nama SFR	Keterangan	Alamat
TH0	Timer 0 High Byte	8Ch
TL0	Timer 0 Low Byte	8Ah
TH1	Timer 1 High Byte	8Dh
TL1	Timer 1 Low Byte	8Bh
TCON	Timer Control	88h
TMOD	Timer Mode	89h

- ❖ TIMER0 memiliki dua macam SFR yang eksklusif bagi dirinya sendiri, yaitu TH0 dan TL0 yang membentuk harga aktual dari timer. Misalnya TIMER0 berharga 1000, maka TH0 akan berisi 3 sedangkan TL0 akan berisi 232. Untuk melihat harga sebenarnya, kalikan harga TH0 dengan 256 dan kemudian tambahkan dengan TL0.
- ❖ TIMER1 identik dengan TIMER0 kecuali bahwa SFR eksklusif yang dimilikinya adalah TH1 dan TL1. Dan karena kedua timer ini memiliki kapasitas dua byte, maka harga maksimum yang bisa ditampung adalah 65.535. Dengan demikian, apabila timer telah melampaui harga 65.535, maka dia akan reset atau overflow dan kemudian kembali ke harga awal 0.

SFR TMOD

SFR TMOD digunakan untuk mengontrol mode operasi dari kedua timer. Setiap bit dari SFR ini menyediakan informasi bagi mikrokontroler bagaimana menjalankan timer. Empat bit orde tinggi (bit 4 hingga bit 7) berhubungan dengan TIMER1, sedangkan empat bit orde bawah (bit 0 hingga bit 3) mempunyai fungsi sama yang diperuntukkan bagi TIMER0.

Daftar Bit SFR TMOD

Bit	Nama	Fungsi	Timer
7	GATE 1	Jika bit ini diset, timer hanya akan bekerja jika INT1 (P3.3) berlogika 1. Jika bit ini dinolkan, timer akan bekerja tanpa dipengaruhi kondisi INT1	1
6	C/T1	Jika bit ini diset, timer akan menghitung kondisi pada T1 (P3.5). Jika bit ini dinolkan, timer akan bertambah satu setiap siklus mesin	1
5	T1M1	Bit mode timer	1
4	T1M0	Bit mode timer	1
3	GATE0	Jika bit ini diset, timer hanya akan bekerja jika INT0 (P3.2) berlogika 1. Jika bit ini dinolkan, timer akan bekerja tanpa dipengaruhi kondisi INT0	0
2	C/T0	Jika bit ini diset, timer akan menghitung kondisi pada T0 (P3.4). Jika bit ini dinolkan, timer akan bertambah satu setiap siklus mesin	0
1	T0M1	Bit mode timer	0
0	T0M0	Bit mode timer	0

Seperti terlihat pada tabel di atas, ada 4 bit yang menyatakan mode untuk kedua timer. Masing- masing dua bit untuk satu timer. Adapun mode operasi yang dimaksud di sini adalah sebagaimana tercantum dalam tabel di bawah ini.

Mode Operasi Timer

TxM1	TxM0	Mode Timer	Keterangan
0	0	0	Timer 13 bit
0	1	1	Timer 16 bit
1	0	2	8 bit auto reload
1	1	3	Mode timer split

Mode Timer 13 Bit (Mode 0)

Dalam mode 0, timer yang dibentuk adalah timer 13 bit. Mode ini digunakan untuk menjaga kompatibilitas pendahulu keluarga 8051, yaitu generasi 8048. Pada saat ini timer 13 bit sudah jarang digunakan. Saat timer diset sebagai timer 13 bit, TLx akan mencacah dari 0 hingga 31. Jika TLx melebihi 31, maka dia akan reset ke harga awal 0 dan kemudian menambah harga THx. Dengan demikian, hanya 13 bit dari dua byte yang digunakan, yaitu bit 0-4 dari TLx dan bit 0-7 dari THx. Sehingga maksimum harga yang bisa dicapai adalah 8.192. Sehingga jika timer diset dalam mode ini, dia akan menjadi nol setelah 8.192 siklus mesin.

Mode Timer 16 Bit (Mode 1)

Timer mode 1 adalah timer 16 bit. Mode ini adalah mode yang paling umum digunakan. Fungsinya sama dengan timer 13 bit, namun yang didayagunakan adalah 16 bit. TLx akan mencacah dari 0 hingga 255. Jika TLx melebihi 255, dia akan reset menjadi 0 dan menambah THx dengan 1. Karena kemampuan 16 bit, maka mode ini memiliki batas maksimum harga 65.535. Sehingga jika timer diset dalam mode ini, dia akan menjadi nol setelah 65.535 siklus mesin.

Mode Timer 8-Bit Auto Reload (Mode 2)

Timer mode 2 adalah timer 8 bit dengan kemampuan pengisian ulang (auto reload). Dalam mode ini, THx akan menyimpan harga awal counter dan TLx berfungsi sebagai timer 8-bit. TLx akan memulai mencacah dengan harga yang tersimpan pada THx, dan jika telah melampaui harga 255, dia akan reset dan kembali ke harga awal yang tersimpan di THx.

Sebagai contoh, dimisalkan TH0 menyimpan harga FDh dan TL0 harga aktualnya FEh, maka untuk beberapa siklus mesin akan didapatkan urutan perubahan harga sebagai mana tercantum dalam tabel di bawah ini.

Contoh Perubahan Harga TL0

Siklus Mesin	Harga TH0	Harga TL0
1	FDh	FEh
2	FDh	FFh
3	FDh	FDh
4	FDh	FEh
5	FDh	FFh
6	FDh	FDh
7	FDh	FEh

Seperti terlihat di atas, harga TH0 tidak pernah berubah. Dengan demikian dalam mode 2, THx merupakan variabel yang menentukan waktu sedangkan TLx adalah timer yang selalu mencacah secara konstan setiap siklus mesin. TLx akan overflow dan reset ke harga yang tersimpan dalam TH0. Keuntungan yang didapatkan karena fleksibilitas penentuan tenggang waktu dengan mengatur harga pada THx.

Contohnya jika diinginkan timer yang selalu menghitung dari 200 hingga 255. Jika digunakan mode 0 atau 1, diperlukan pengecekan terus menerus apakah timer mengalami overflow atau tidak. Dan jika benar terjadi overflow, maka diperlukan kode untuk mereset timer ke harga 200. Hal ini akan memerlukan banyak instruksi dan memakan waktu sehingga tidak efisien. Namun, jika digunakan mode 2, tidak diperlukan monitor terus-menerus terhadap kondisi timer. Cukup masukkan harga 200 pada THx dan biarkan mikrokontroler yang mengatur agar timer selalu mencacah dari 200 hingga 255.

Mode Timer Split (Mode 3)

Timer mode 3 adalah mode timer split. Jika TIMER0 diset dalam mode 3, dia akan menjadi dua timer 8 bit yang berbeda. Timer 0 adalah TLO dan Timer 1 adalah TH0. Kedua-duanya akan mencacah dari 0 hingga 255 dan jika menemui kondisi overflow akan reset ke nol. Saat TIMER0 dalam mode split, TIMER1 bisa diset pada mode 0, 1, atau bahkan 2 secara normal. Mode ini hanya dipakai jika diperlukan dua timer 8 bit yang terpisah.

SFR TCON

SFR ini mengontrol kedua timer dan menyediakan informasi yang sangat berguna berkaitan dengan timer-timer tersebut. Struktur SFR TCON dapat dilihat pada tabel di bawah ini.

Bit-bit SFR TCON

Bit	Nama	Alamat	Fungsi	Timer
7	TF1	8Fh	Timer 1 Overflow. Bit ini diset oleh mikrokontroler jika Timer 1 overflow	1
6	TR1	8Eh	Timer 1 Run. Jika bit ini diset maka Timer 1 akan bekerja. Sebaliknya jika direset maka Timer 1 akan mati	1
5	TF0	8Dh	Timer 0 Overflow. Bit ini diset oleh mikrokontroler jika Timer 0 overflow	0
4	TR0	8Ch	Timer 0 Run. Jika bit ini diset maka Timer 0 akan bekerja. Sebaliknya jika direset maka Timer 1 akan mati	0

Dalam tabel hanya dicantumkan 4 bit dari 8 bit yang ada pada SFR TCON. Hal ini karena hanya 4 bit (bit 4 hingga bit 7) yang berkaitan dengan timer, sedangkan bit sisanya berkaitan dengan interupsi yang akan dibahas selanjutnya. Untuk mengeset atau mereset bit-bit SFR tidak perlu dengan memberikan nilai 8 bit. Bit-bit SFR bisa dialamati per bit. Dengan demikian perubahan satu atau beberapa bit tidak akan mengganggu status bit-bit yang lain.

Membaca Status Timer

Membaca status timer ada dua cara. Yang pertama dengan membaca harga aktual 16 bit dari timer, dan yang kedua adalah mendeteksi apakah timer menemui kondisi overflow. Jika timer yang digunakan adalah timer mode 8-bit, pembacaan harga aktual cukup mudah. Bacalah harga 1 byte tersebut dan selesailah sudah.

Namun jika timer yang digunakan adalah mode 13 bit atau 16 bit, permasalahan menjadi lebih rumit. Bagaimana jika harga aktual low byte adalah 255 dan pembacaan high byte adalah 15. Seharusnya harga sebenarnya adalah high byte 14 dan low byte adalah 255, karena saat membaca low byte sebesar 255, beberapa saat kemudian high byte akan bertambah satu saat pembacaan, sehingga pembacaan menjadi meleset sebesar 256 hitungan karena terletak pada high byte.

Pemecahannya adalah dengan membaca high byte terlebih dahulu dan kemudian membaca low byte. Setelah itu high byte dibaca lagi dan kemudian dibandingkan dengan pembacaan semula, bila berbeda, maka yang dipakai adalah pembacaan high byte yang pertama.

Kadang yang perlu diketahui hanyalah saat timer reset menjadi nol. Dengan kata lain, tidak penting berapa harga aktual dari timer, namun kapan timer overflow dan kembali menjadi nol. Saat overflow, mikrokontroler secara otomatis mengeset bit TFX dalam register TCON. Ini berarti pengecekan overflow cukup dengan mengecek apakah bit TFX set atau tidak. Dengan cara ini bisa dibuat program untuk menentukan selang yang pasti. Dari pembahasan sebelumnya, diketahui bahwa untuk mendapatkan selang 0,05 detik diperlukan pencacahan hingga 46.079 kali. Penggunaan mode ini akan menginisialisasi timer dengan harga selisih antara 65.535 dan 46.079, yaitu 19.457. Sehingga 46.079 cacahan berikutnya setelah 19.457 akan menyebabkan timer overflow.

Segmen Program Contoh Timer

```
MOV TH0,#76      ; (76X256=19.456)
MOV TL0,#01      ; (19.456+1=19.457)
MOV TMOD,#01     ; Timer 0 Mode 16-bit
SETB TR0         ; Start Timer 0
JNB TF0,$        ; Loop sampai overflow
```

Timer Sebagai Penghitung Kejadian

Di atas telah dibahas mengenai timer sebagai penghitung selang waktu. Namun 89C51 juga memungkinkan timer digunakan sebagai penghitung (counter). Sebagai contoh, sebuah sensor dipasang di jalan untuk menghitung jumlah mobil yang lewat. Setiap mobil lewat, dia akan mengirim pulsa yang dapat dihubungkan dengan kaki 89C51 sebagai monitor.

Jika digunakan TIMER0 untuk menghitung jumlah mobil yang lewat, bit C/T0 yang dimanfaatkan. Jika bit C/T0 ini diset maka TIMER0 akan memonitor kaki P3.4. Sehingga jika jumlah mobil aktual yang ingin diketahui, cukup dengan membaca harga yang tersimpan dalam TIMER0.

M. Interupsi

Seperti namanya, interupsi adalah suatu kejadian yang akan menghentikan sementara jalan program saat itu. Dengan interupsi, suatu alur program dapat dihentikan sementara untuk menjalankan suatu subrutin, dan kemudian melanjutkan aliran program secara normal seperti tidak pernah ada interupsi. Subrutin ini yang disebut dengan interrupt handler, dan hanya dijalankan jika terjadi suatu kejadian khusus (event). Kejadian ini bisa berupa timer yang mengalami overflow, penerimaan karakter melalui port serial, mengirimkan karakter melalui port serial, atau salah satu dari dua kejadian eksternal. Mikrokontroler 89C51 bisa dikonfigurasi untuk menangani interupsi yang disebabkan oleh salah satu dari kejadian.

Dengan interupsi ini, dapat dengan mudah dimonitor kejadian-kejadian yang diinginkan. Tanpa interupsi maka proses monitor ini dilakukan manual dengan pengecekan berulang. Proses pengecekan manual ini akan membuat program menjadi panjang dan lebih rumit.

Kejadian-kejadian yang Membangkitkan Interupsi

Dalam keluarga 8051, ada beberapa kejadian yang dapat membangkitkan interupsi, yaitu:

- a) Timer 0 overflow
- b) Timer 1 overflow
- c) Penerimaan / Pengiriman data secara serial
- d) Kejadian eksternal 0
- e) Kejadian eksternal 1

Dengan demikian, perlu dibedakan interupsi yang terjadi karena kejadian-kejadian yang berbeda. Hal ini bisa diwujudkan dengan adanya alamat pasti bagi sebuah rutin interupsi untuk masing-masing interupsi.

Daftar Alamat Rutin Interupsi

Interupsi	Flag	Alamat Rutin
Eksternal 0	IE0	0003h
Timer 0	TF0	000Bh
Eksternal 1	IE1	0013h
Timer 1	TF1	001Bh
Serial	R1/T1	0023h

Dengan referensi tabel di atas, dapat dilihat bahwa jika TIMERO mengalami overflow (bit TF0 logika 1), maka program utama akan ditunda sementara dan kontrol program akan melompat ke alamat 0003h.

Menyeting Interupsi

Pada saat awal dihidupkan (power on), semua interupsi dimatikan. Jadi, meskipun bit TF0 diset (interupsi timer 0 diaktifkan), mikrokontroler 89C51 tidak akan menjalankan rutin interupsi. Sebuah program harus dibuat dan dijalankan untuk mengaktifkan dan menspesifikasikan interupsi mana yang diinginkan. Hal ini dapat dilakukan dengan mengkonfigurasi SFR IE (Interrupt Enable) yang berada di alamat A8h. Konfigurasi bit-bit SFR IE yang mengatur enable dari konfigurasi interupsi tercantum pada tabel di bawah ini.

Konfigurasi Bit-bit SFR IE

Bit	Nama	Alamat	Fungsi
7	EA	AFh	Global enable/disable interupsi
6	-	AEh	Tidak didefinisikan
5	-	ADh	Tidak didefinisikan
4	ES	ACh	Enable interupsi serial
3	ET1	ABh	Enable interupsi timer 1
2	EX1	AAh	Enable interupsi eksternal 1
1	ET0	A9h	Enable interupsi timer 0
0	EX0	A8h	Enable interupsi eksternal 0

Sebelum semua interupsi dijalankan, bit 7 dari SFR IE harus diset terlebih dahulu. Global enable/ disable interupsi ini menyebabkan semua interupsi enable atau disable. Jika bit 7 IE ini dinolkan, tidak akan ada satu interupsi pun yang terjadi. Prosedur yang benar adalah dengan mengeset jenis interupsi yang diinginkan dan kemudian mengeset bit 7 IE, barulah interupsi akan berjalan dengan baik.

Prioritas Interupsi

Mikrokontroler 89C51 menawarkan dua macam prioritas interupsi, yaitu prioritas tinggi dan prioritas rendah. Dengan demikian, dapat ditentukan suatu interupsi memiliki prioritas pengerjaan lebih dahulu daripada interupsi-interupsi yang lain yang bekerja saat itu. Sebagai contoh, interupsi timer 1 diaktifkan untuk secara otomatis memanggil sebuah rutin saat timer 1 mengalami kondisi overflow. Di samping itu, interupsi serial juga diaktifkan untuk memanggil suatu rutin saat ada sebuah karakter diterima dalam port serial. Dalam kasus ini diinginkan bahwa penerimaan karakter dari port serial lebih penting daripada interupsi timer. Jadi seandainya interupsi timer 1 sedang bekerja dan terjadi

penerimaan karakter, maka rutin interupsi timer 1 tersebut akan diinterupsi oleh interupsi port serial dan akan ditunda sementara untuk mengerjakan rutin interupsi port serial ini. Setelah pengerjaan rutin interupsi serial selesai, maka kontrol program akan kembali ke rutin interupsi timer 1. Yang dikerjakan di sini adalah mengeset prioritas interupsi serial lebih tinggi daripada prioritas interupsi timer. Prioritas interupsi ini dikontrol oleh SFR IP dengan alamat B8h yang memiliki konfigurasi bit-bit sebagaimana tercantum dalam tabel di bawah ini.

Konfigurasi Bit-bit SFR IP

Bit	Nama	Alamat	Fungsi
7	-		Tidak terdefinisi
6	-		Tidak terdefinisi
5	-		Tidak terdefinisi
4	PS	BCh	Prioritas interupsi serial
3	PT1	BBh	Prioritas interupsi timer 1
2	PX1	BAh	Prioritas interupsi eksternal 1
1	PT0	B9h	Prioritas interupsi timer 0
0	PX0	B8h	Prioritas interupsi eksternal 0

Pemakaian prioritas interupsi di atas memiliki beberapa peraturan yang tercantum di bawah ini:

- (a). Tidak ada interupsi yang meninterupsi interupsi prioritas tinggi.
- (b). Interupsi prioritas tinggi boleh menginterupsi interupsi prioritas rendah.
- (c). Interupsi prioritas rendah boleh terjadi jika tidak ada interupsi lain yang sedang dijalankan.
- (d). Jika dua interupsi terjadi pada waktu bersamaan, interupsi yang memiliki prioritas lebih tinggi akan dikerjakan terlebih dahulu. Jika keduanya memiliki prioritas sama, maka interupsi yang berada pada urutan polling akan dikerjakan terlebih dahulu.

Mikrokontroler 89C51 secara otomatis akan menguji apakah sebuah interupsi bisa terjadi setelah setiap instruksi dikerjakan. Pengecekan ini mengikuti suatu alur yang disebut dengan Polling Sequence dengan urutan:

- Interupsi Eksternal 0
- Interupsi Timer 0
- Interupsi Eksternal 1
- Interupsi Timer 1
- Interupsi serial

Ini berarti jika sebuah interupsi serial terjadi pada waktu bersamaan dengan interupsi eksternal 0, maka interupsi eksternal 0 akan dikerjakan terlebih dahulu dan interupsi serial baru akan dikerjakan setelah pengerjaan rutin interupsi eksternal 0 selesai dilakukan.