



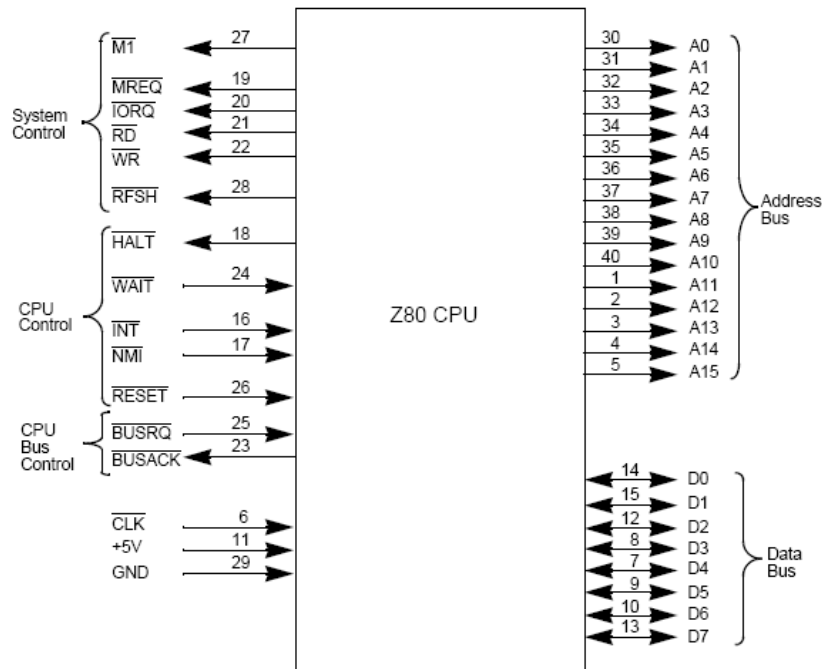
## KONFIGURASI PIN-PIN MIKROPROSESOR Z 80

Yoyo somantri  
Dosen Jurusan Pendidikan Teknik Elektro  
FPTK Universitas Pendidikan Indonesia

### 1. Konfigurasi

Pengelompokan pin – pin dari mikroprosesor Z80 dapat dilihat pada gambar 1. Komponen - komponen dalam sistem mikroprosesor dapat melakukan komunikasi satu sama lain melalui saluran komunikasi yang disebut bus. Pada mikroprosesor Z80 terdapat 3 jenis bus, yaitu :

- Bus data (*Data Bus*)
- Bus alamat (*Address Bus*)
- Bus kontrol (*Control Bus*)



Gambar 1. Konfigurasi pin Z80  
(Sumber dari data book mikroprosesor Z80)

Bus data pada mikroprosesor Z80 mempunyai 8 bit, digunakan untuk mengirim atau menerima data antara komponen – komponen sistem dengan mikroprosesor. Bus

data merupakan *bidirectional* bus (bus 2 arah) artinya dapat menerima atau mengirim data.

Bus alamat pada mikroprosesor Z80 berjumlah 16 bit, sehingga dapat menghubungkan 64 Kb memori. Bus alamat ini merupakan bus satu arah yang digunakan untuk mengirim alamat lokasi memori.

Bus kontrol pada mikroprosesor Z80 ada 3 jenis, yaitu :

1. Sinyal control system :

- a.  $\overline{M1}$  : merupakan sinyal *control* keluaran. Bila sinyal ini berlogika 0, menunjukkan bahwa siklus yang sedang berlangsung adalah siklus pengambilan kode operasi (*op-code*) dalam pelaksanaan suatu instruksi.
- b.  $\overline{MREQ}$  (*Memory Request*) : merupakan sinyal control keluaran. Bila sinyal ini berlogika 0, menunjukkan bahwa bus alamat berisi alamat memori untuk pembacaan atau penulisan memori atau komponen I/O.
- c.  $\overline{IORQ}$  (*Input/ Output Request*) : merupakan sinyal control keluaran. Bila sinyal ini berlogika 0, berarti bus alamat A0 – A7 berisi alamat komponen I/O yang dipilih sedangkan A8 – A15 dapat diisi data dari akumulator.
- d.  $\overline{RD}$  (*Read*) : merupakan sinyal control keluaran. Bila sinyal ini berlogika 0, berarti sedang dilakukan pembacaan data dari memori atau komponen I/O.
- e.  $\overline{WR}$  (*Write*) : merupakan sinyal control keluaran. Bila sinyal ini berlogika 0, berarti sedang dilakukan operasi penulisan data ke memori atau komponen I/O.
- f.  $\overline{RFSH}$  (*Refresh*) : merupakan sinyal control keluaran. Bila sinyal ini berlogika 0, berarti 8 bit terbawah bus alamat berisi alamat dinamik (yang ditunjuk oleh register R) yang harus di *refresh*.

2. Sinyal control CPU :

- a.  $\overline{HALT}$  : merupakan sinyal kontrol keluaran. Bila sinyal ini berlogika 0, berarti suatu instruksi HALT sedang dilaksanakan. CPU melaksanakan instruksi NOP (*No Operation*) dan menunggu instruksi NMI.

- b.  $\overline{WAIT}$  : merupakan sinyal control masukan. Bila sinyal ini berlogika 0, menunjukkan bahwa data dari memori atau I/O tidak siap untuk ditransfer ke bus data.
- c.  $\overline{INT}$  (Interrupt) : merupakan sinyal control masukan. Bila sinyal ini berlogika 0, berarti suatu alat luar meminta suatu interupsi. CPU akan menahan permintaan sampai pada akhir siklus intruksi, jika interupsi *flip-flop* di *set* dan jika sinyal  $BUSRQ$  tidak aktif.
- d.  $\overline{NMI}$  (*Non Maskable Interrupt*) : merupakan saluran kontrol masukan. Merupakan suatu interupsi yang tidak dapat dihalangi dengan instruksi – instruksi *software*. Sinyal ini aktif jika berlogika 0.
- e.  $\overline{RESET}$  : merupakan sinyal control masukan. Bila sinyal ini berlogika 0 digunakan untuk me-reset CPU yang berarti PC diisi dengan 0000H dan *interrupt flip – flop* (IFF) di *disable*, *Register I* diset ke 00H, interupsi diset pada mode 0. Selama waktu *reset*, bus alamat dan bus data menuju ke keadaan impedansi tinggi dan sinyal keluaran kontrol berada dalam keadaan tidak aktif.

### 3. Sinyal control bus :

- a.  $\overline{BUSRQ}$  (*Bus Request*) : merupakan sinyal kontrol masukan. Sinyal ini adalah sinyal permintaan bus, digunakan untuk meminta bus data, bus alamat dan bus control CPU menuju ke keadaan impedansi tinggi sehingga peralatan lain dapat mengontrol bus-bus ini.
- b.  $\overline{BUSAK}$  (*Bus Acknowledge*) : merupakan sinyal kontrol keluaran. Menyatakan bahwa *bus data*, *bus alamat* dan *bus kontrol* sudah diset pada keadaan impedansi tinggi dan peralatan luar dapat mengontrol bus-bus.

### 3. Memperkenalkan mikroprosesor keluarga Intel

Mikroprosesor produksi Intel awal terkenalnya dimulai dari mikroprosesor 8080 dan 8085 yang mempunyai jalur data 8 bit. Sedangkan mikroprosesor yang dipergunakan pada *Personal computer* (PC) dimulai dari  $\mu P$  8086/8088 yang mempunyai jalur data 16 bit dan mampu mengalami memori sebesar 1 M byte

atau 1.048.576 byte.  $\mu$ P 80286 juga termasuk  $\mu$ P berarsitektur internal 16 bit tetapi mampu mengamati 16 M byte dengan kecepatan lebih tinggi dari mikroprosesor sebelumnya.

Mikroprosesor 80386, 80486, Pentium I, II, III, dan Pentium 4 mempunyai arsitektur internal 32 bit dan pemrogramannya kompatibel dengan mikroprosesor keluarga Intel sebelumnya. Pada pemrograman ada model pemrograman *visible* dan pemrograman *invisible*. Program *visible* terdapat pada mikroprosesor dari 8086 sampai Pentium 4. Sedangkan program *invisible* hanya pada mikroprosesor 80286 sampai Pentium 4.

Mikroprosesor Intel 80386 dapat beroperasi pada mode riil, maksudnya mikroprosesor tersebut dapat dioperasikan sebagai mikroprosesor 8088 atau 8086 dengan kecepatan jauh lebih cepat. Memori dapat dijangkau juga hanya 1 M byte sama seperti mikroprosesor 8088.  $\mu$ P 80386 juga dapat beroperasi dengan mode proteksi, maksudnya dapat beroperasi seperti  $\mu$ P 80286 dengan jangkauan memori 16 M byte dengan bekerja lebih cepat.

$\mu$ P 80386 Intel dapat bekerja dengan mode Virtual, maksudnya mikroprosesor bekerja utuh pada 32 bit data. Memori yang dapat dijangkau tidak hanya memori riil yang ada di RAM akan tetapi juga memori Virtual yang menggunakan *harddisk* atau media lainnya yang cepat dan berkapasitas besar.

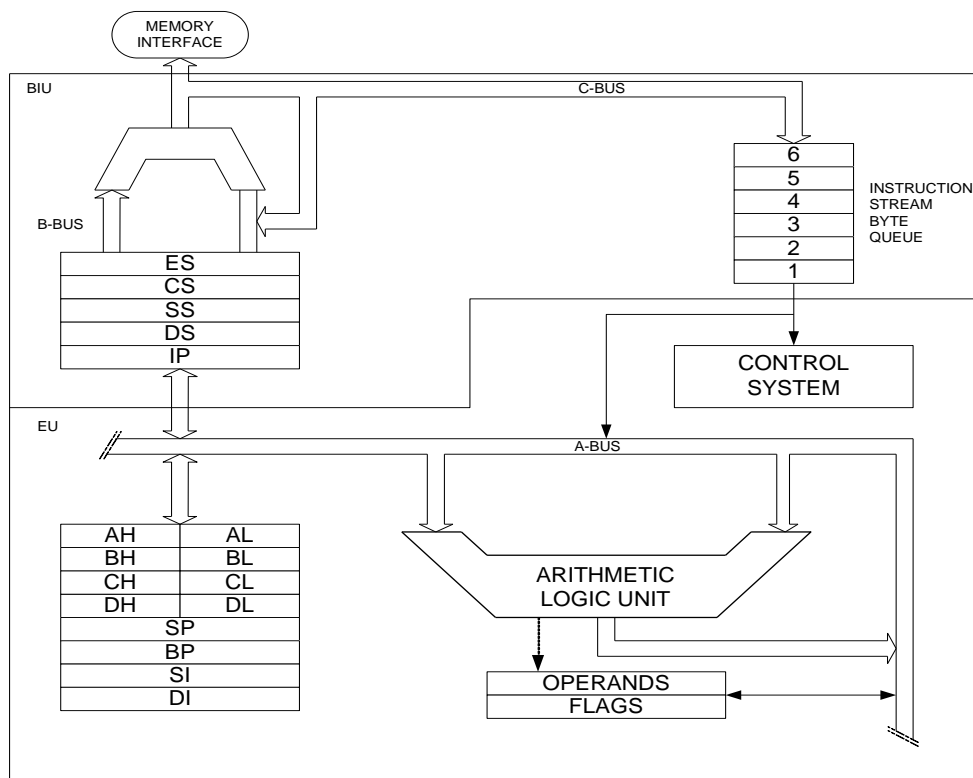
Mikroprosesor Pentium I, II, III, dan Pentium 4 pada umumnya mempunyai lebar jalur data 64 bit dengan ukuran memori mulai dari 4 G byte sampai dengan 64 G byte ditambah dengan cache dengan berbagai ukuran dimulai dari 8 K, 16 K, 32 K, 256 K, dan 512 K cache. Bahkan sedang direncanakan untuk mengeluarkan arsitektur mikroprosesor baru dengan mempunyai lebar jalur data 128 bit. Arsitektur ini yang diberi nama kode "*Merced*" merupakan *Joint venture* yang disebut EPIC (*Explicitly Parallel Instruction Computing*) antara Intel dengan Hewlett-Packard. Pada mikroprosesor ini perubahan yang meliputi 128 *register* multiguna, 128 *register floating point*, *register predicate* dan beberapa unit eksekusi.

*Register-register* yang dipergunakan untuk pemrograman, yaitu *register* 8 bit antara lain *register* AH, AL, BH, BL, CH, CL, DH, dan DL, pada umumnya terdapat pada mikroprosesor 8085, 8086, 8088, 8051, dan 8048. *Register* 16 bit, yaitu

register AX, BX, CX, DX, SP, BP, DI, SI, IP, FLAGS, CS, DS, ES, SS, FS, dan GS yang dimiliki oleh mikroprosesor 80186, 80286, 80386 SL, 80386 SX, dan 80386 EX. Sedangkan register 32 bit, yaitu register EAX, EBX, ECX, EDX, ESP, EBP, EDI, ESI, EIP, dan EFLAG yang terdapat pada mikroprosesor 80386 DX, 80486, Pentium I, II, III, dan 4.

Secara konseptual disain arsitektur mikroprosesor mulai dari 80486 adanya unit koprosesor dan cache. Sedangkan untuk mikroprosesor Pentium disain arsitekturnya mengintegrasikan beberapa CPU, koprosesor dan cache.

Dibawah ini diperkenalkan arsitektur mikroprosesor 8086/8088 pada gambar 3



Gambar 3. Arsitektur mikroprosesor 8086/8088 Intel.

(Sumber dari data book Mikroprosesor Intel, tahun1985)

Pada  $\mu P$  8086/8088 registernya dapat dikelompokkan menjadi beberapa kelompok antara lain:

- 1) *Register data* disebut juga *general purpose register*.

- a) Akumulator atau *register* AX, yaitu AH dan AL masing-masing 8 bit digunakan untuk melakukan operasi *arithmetic*, logika, *shift*, *rotate*, dan lintas data ke dan dari *port*.
- b) *Base register* BX, yaitu BH dan BL masing-masing 8 bit digunakan sebagai pencatat alamat memori yang isinya akan dilaksanakan disamping itu juga dapat dioperasikan *arithmetic*, logika, *shift*, dan *rotate*.
- c) *Counter register* CX, yaitu CH dan CL masing-masing 8 bit digunakan sebagai penghitung *register* terhadap perintah *loop* dan perintah blok *transfer* yang termasuk operasi *string*.
- d) *Data Register* DX, yaitu DH dan DL masing-masing 8 bit digunakan sebagai pencatat alamat *port* secara khusus, baik alamat *port* 8 bit ataupun alamat *port* 16 bit.

## 2) *Register Index* dan *Pointer*.

- a) *Stack Pointer* SP (16 bit), digunakan untuk mencatat alamat *stack*. *Stack* adalah memori khusus yang digunakan untuk menyimpan isi *register* yang menampung perintah *Push* dan *Pop*.
- b) *Base Pointer* BP (16 bit), digunakan untuk mencatat alamat data yang disimpan di *stack*, bila mana data tersebut akan digarap.
- c) *Register* SI dan DI (16 bit) berfungsi sebagai *destinasi index register*. Kedua *register* ini mampu mencatat alamat memori yang isi memorinya akan digarap. Penunjukkan alamat memori menggunakan SI dan DI dapat menggunakan selisih alamat terhadap alamat basis dengan rumus isi SI  $\pm d$  dimana d adalah selisih alamat berupa angka – 128 sampai + 127.

## 3) *Register Segment* terdiri dari :

- a) *Code segment* atau *register* CS (16 bit) untk mencatat alamat segmen memori tempat kode operasi (*opcode*) suatu program dimana kode itu secara berurut harus dibaca dan dilaksanakan oleh CPU.
- b) *Data segment* atau *register* DS (16 bit) untuk mencatat alamat segmen memori tempat penyimpanan data-data di memori yang digunakan oleh suatu program.
- c) *Stack segment* atau *register* SS (16 bit) digunakan untuk mencatat alamat segmen memori yang dipergunakan menjadi *stack* (tumpukan). *Stack* adalah memori yang dipilih digunakan untuk menyimpan dan membaca dengan operasi *Push* dan *Pop*.

d) *Extra Segment* atau *register AS* (16 bit) digunakan untuk mencatat alamat segment memori bagi data yang tidak terletak di *code segment*, tidak terletak di *data segment*, ataupun tidak terletak pada *stack segment*, misalnya pada *video display* dan lain sebagainya.

4) *Register* pelengkap yang terdiri dari :

a) *Register Flag* (16 bit) . 9 bit dari *register* tersebut digunakan untuk mencatat tanda, kondisi kerja mikroprosesor yaitu : *overflow flag* (OF), *direct flag* (DF), *bit interrupt flag* (IF), dan *bit trap flag* (TF). Sedangkan tanda untuk operasi *arithmetic* dan *logic* terdiri dari : *bit sign flag* (SF), *Zero flag* (ZF), *auxiliary carry flag* (AF), *parity flag* (PF), dan *carry flag* (CF). Dengan format sebagai berikut :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

	X	X	X	X	O	D	IF	T	S	Z	X	A	X	P	X	C
Tanda					F	F		F	F	F		F		F		F
X tidak dipergunakan.																

b) *Instruction Pointer* atau *register IP* (16 bit), yaitu *register* bersama dengan *register* kode segmen mencatat alamat kode-kode bahas mesin suatu program yang berada pada RAM. Setiap program terdiri dari sejumlah instruksi.

Tempat kode mesin (*opcode*) pada RAM alamatnya dicatat oleh *Code Segment* dan *Instruction Pointer* ( CS : IP ). Bila program akan dieksekusi maka *code segment* dan *instruction pointer* harus mencatat alamat awal letak permulaan kode bahasa mesin tersebut. Jika program dilaksanakan berarti CS : IP menaikkan isinya satu demi satu sehingga secara teratur maka kode bahasa mesin dibaca oleh prosesor. *Program counter* atau *instruction pointer*, bila program selesai maka CS:IP mencatat lokasi terakhir setelah kode bahasa mesin terakhir. Oleh karena itu kode segmen dan *instruction Pointer* berfungsi sebagai *Program Counter* dan ditulis dengan format CS: IP.