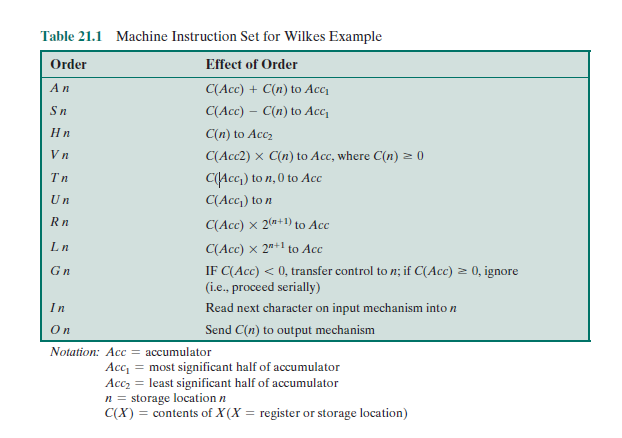
Istilah micro-program pertama kali muncul padatahun 1950 dan diperkenalkan oleh M. V. Wilkes Pada hard-wired diperlukan perancangan ulang perangkat keras jika serangkaian instruksi

dikembangkan. Sebaliknya, pada micro-programmed, serangkaian instruksi mikro (program-mikro) disimpan dalam ROM sehingga sebuah instruksi dapat diubah dengan mengubah program mikro yang bersesuaian dengan instruksi tersebut dan kelompok instruksi dapat dikembangkan dengan hanya menyertakan ROM tambahan yang berisi program-mikro bersesuaian.

CHAPTER 21.1 Basic Concepts

**Microinstructions**

Unit kontrol tampaknya perangkat yang cukup sederhana. Namun demikian, untuk menerapkan unit kontrol sebagai interkoneksi elemen logika dasar bukanlah tugas yang mudah. Rancangan harus mencakup logika untuk pengurutan melalui operasi mikro, untuk menjalankan operasi mikro, untuk menafsirkan opcode, dan untuk membuat keputusan berdasarkan bendera ALU. Sulit untuk merancang dan menguji perangkat keras seperti itu. Selanjutnya, desainnya relatif tidak fleksibel. Misalnya, sulit untuk mengubah desain jika ingin menambahkan instruksi mesin baru.



**Wilkes Control**

Seperti telah disebutkan, Wilkes pertama kali mengusulkan penggunaan unit kontrol mikroprogram pada tahun 1951 [WILK51]. Proposal ini kemudian diuraikan menjadi desain yang lebih rinci [WILK53]. Ini adalah pelajaran untuk memeriksa proposal seminal ini. Konfigurasi yang diusulkan oleh Wilkes digambarkan pada Gambar 21.5. Inti dari sistem adalah matriks yang sebagian diisi dengan dioda. Selama siklus mesin, satu baris matriks diaktifkan dengan pulsa. Ini menghasilkan sinyal pada titik-titik di mana ada dioda (ditunjukkan oleh titik di diagram). Bagian pertama dari baris menghasilkan sinyal kontrol yang mengontrol pengoperasian prosesor. Bagian kedua menghasilkan alamat baris yang akan berdenyut di siklus mesin berikutnya. Dengan demikian, setiap baris dari matriks adalah satu microinstruction, dan tata letak matriks adalah memori kontrol.

Pada awal siklus, alamat baris yang akan di-pulsed terdapat di Register I. Alamat ini adalah input ke decoder, yang ketika diaktifkan oleh pulsa clock, mengaktifkan satu baris dari matriks. Tergantung pada sinyal kontrol, baik opcode dalam register instruksi atau bagian kedua dari baris yang digetarkan dilewatkan ke Register II selama siklus. Register II kemudian dikunci ke Register I oleh pulsa clock. Bolak pulsa clock digunakan untuk mengaktifkan baris matriks dan untuk mentransfer dari Register II ke Register I. Pengaturan dua-register diperlukan karena decoder hanyalah rangkaian kombinatorial; hanya dengan satu register, output akan menjadi input selama siklus, menyebabkan kondisi yang tidak stabil.

**Advantages and Disadvantages**

Keuntungan utama dari penggunaan microprogramming untuk menerapkan kontrolunit adalah menyederhanakan desain unit kontrol. Dengan demikian, keduanya lebih murah dankesalahan kurang rawan untuk menerapkan. Unit kontrol yang dipasok harus berisi logika yang kompleksuntuk pengurutan melalui banyak operasi mikro dari siklus instruksi. Di atassisi lain, decoder dan unit logika sekuensing dari kontrol microprogrammedunit adalah potongan logika yang sangat sederhana.

CHAPTER 21.2 MICROINSTRUCTION SEQUENCING

Dua tugas dasar yang dilakukan oleh unit kontrol microprogrammed adalah sebagai berikut:

■**Microinstruction sequencing**: Dapatkan microinstruction berikutnya dari kontrol

ingatan.

■**Microinstruction execution**: Menghasilkan sinyal kontrol yang diperlukan untuk mengeksekusi

microinstruction.

Dalam mendesain unit kontrol, tugas-tugas ini harus dipertimbangkan bersama-sama, karena

keduanya mempengaruhi format microinstruction dan timing dari unit kontrol. Dibagian ini, kami akan fokus pada urutan dan mengatakan sesedikit mungkin tentang formatdan masalah waktu. Masalah-masalah ini diperiksa secara lebih rinci di bagian selanjutnya.

**Design Considerations**

Dua kekhawatiran terlibat dalam desain teknik sekuensing microinstruction:ukuran dari microinstruction dan waktu generasi penerus. Perhatian pertamasudah jelas; meminimalkan ukuran memori kontrol mengurangi biaya komponen itu.Perhatian kedua adalah keinginan untuk mengeksekusi microinstructions dengan cepatmungkin.

Dalam menjalankan mikroprogram, alamat dari instruksi mikro berikutnya

dijalankan dalam salah satu kategori ini:

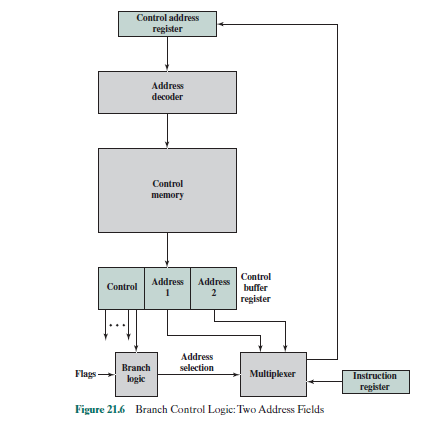
1. Determined by instruction register

2. Next sequential address

3. Branch

Kategori pertama hanya terjadi sekali per siklus instruksi, tepat setelah instruksi diambil.Kategori kedua adalah yang paling umum di sebagian besar desain. Namun, itu desain tidak dapatdioptimalkan hanya untuk akses berurutan. Cabang, keduanya bersyarat dan tanpa syarat, adalah bagianpenting dari mikroprogram. Selanjutnya, microinstruction urutan cenderung pendek; satu dari setiap tigaatau empat microinstructions bisa menjadi cabang [SIEW82]. Dengan demikian, penting untukmerancang kompak, hemat waktu teknik untuk percabangan microinstruction.

Berdasarkan microinstruction saat ini, bendera kondisi, dan isi register instruksi, alamat memori kontrol harus dihasilkan untuk microinstruction berikutnya. Berbagai macam teknik telah digunakan. Kami dapat mengelompokkan mereka ke dalam tiga kategori umum, seperti yang diilustrasikan pada Gambar 21.6 hingga 21.8. Kategori-kategori ini didasarkan pada format informasi alamat di microinstruction:



Pendekatan yang paling sederhana adalah menyediakan dua bidang alamat di setiap microinstruction.Gambar 21.6 menyarankan bagaimana informasi ini akan digunakan. Multiplexer disediakanyang berfungsi sebagai tujuan untuk kedua bidang alamat plus register instruksi.Berdasarkan input alamat-pilihan, multiplexer mentransmisikan opcodetau salah satu dari dua alamat ke register alamat kontrol (CAR). Mobil itu adalahselanjutnya diterjemahkan untuk menghasilkan alamat microinstruction berikutnya. Pemilihan alamatsinyal disediakan oleh modul logika cabang yang inputnya terdiri dari kontrolunit bendera plus bit dari bagian kontrol dari microinstruction.

**Address Generation**

Kami telah melihat masalah sequencing dari sudut pandang pertimbangan formatdan persyaratan logika umum. Sudut pandang lain adalah mempertimbangkanberbagai cara di mana alamat berikutnya dapat diturunkan atau dihitung.Tabel 21.3 daftar berbagai teknik pembangkitan alamat. Ini bisa terjadidibagi menjadi teknik eksplisit, di mana alamat secara eksplisit tersedia dimicroinstruction, dan teknik implisit, yang membutuhkan logika tambahan untuk menghasilkanalamat.

Kami pada dasarnya berurusan dengan teknik eksplisit. Dengan dua bidang

pendekatan, dua alamat alternatif tersedia dengan masing-masing microinstruction. Menggunakan

baik bidang alamat tunggal atau format variabel, berbagai instruksi cabang dapat

diimplementasikan. Instruksi cabang bersyarat tergantung pada jenis berikut

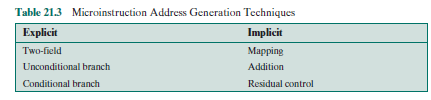
informasi:

■■ bendera ALU

■■ Bagian dari bidang mode opcode atau alamat dari instruksi mesin

■■ Bagian dari daftar yang dipilih, seperti bit tanda

■■ bit Status dalam unit kontrol



**LSI-11 Microinstruction Sequencing**

The LSI-11 is a microcomputer version of a PDP-11, with the main components of

the system residing on a single board. The LSI-11 is implemented using a microprogrammed

control unit [SEBE76].

The LSI-11 makes use of a 22-bit microinstruction and a control memory of

2K 22-bit words. The next microinstruction address is determined in one of five

ways:

* **Next sequential address:** In the absence of other instructions, the control unit’s

control address register is incremented by 1.

* **Opcode mapping:** At the beginning of each instruction cycle, the next microinstruction

address is determined by the opcode.

* **Subroutine facility:** Explained presently.
* **Interrupt testing:** Certain microinstructions specify a test for interrupts. If an

interrupt has occurred, this determines the next microinstruction address.

* **Branch:** Conditional and unconditional branch microinstructions are used.

**21.3 MICROINSTRUCTION EXECUTION**

Siklus microinstruction adalah peristiwa dasar pada prosesor microprogram. Setiap

siklus terdiri dari dua bagian: ambil dan laksanakan. Bagian ambil ditentukan olehgenerasi alamat microinstruction, dan ini ditangani di sebelumnyabagian. Bagian ini berkaitan dengan pelaksanaan suatu instruksi mikro.

Ingat bahwa efek dari pelaksanaan suatu instruksi mikro adalah untuk menghasilkan

sinyal kontrol. Beberapa titik kontrol sinyal ini internal ke prosesor. Itusinyal yang tersisa pergi ke bus kontrol eksternal atau antarmuka eksternal lainnya. Sebagai sebuahfungsi insidental, alamatmicroinstruction berikutnya ditentukan.

**A Taxonomy of Microinstructions**

Microinstructions dapat diklasifikasikan dalam berbagai cara. Perbedaan yang umum

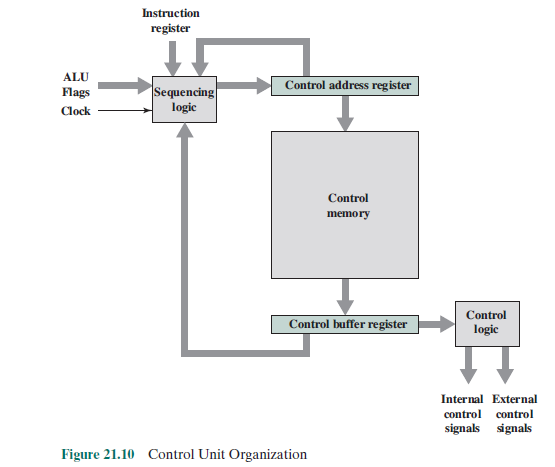
dibuat dalam literatur termasuk yang berikut:

* Vertikal / horizontal
* Dikemas / dibongkar
* microprogramming Keras / lunak
* Enkode langsung / tidak langsung

Semua ini bergantung pada format microinstruction. Tak satu pun dari istilah-istilah ini

telah digunakan secara konsisten, tepat dalam literatur. Namun, pemeriksaandari pasangan-pasangan kualitas ini berfungsi untuk menjelaskan alternatif desain microinstruction.

Dalam paragraf berikut, pertama-tama kita melihat masalah desain utama yang mendasarinyasemua pasangan karakteristik ini, dan kemudian kita melihat konsep yang disarankanoleh masing-masing pasangan.



Berdasarkan sebelumnya, kita dapat melihat bahwa bagian sinyal kontrol dariformat microinstruction jatuh pada spektrum. Pada satu ekstrim, ada satu bit untuksetiap sinyal kontrol; pada ekstrim lainnya, format yang sangat dikodekan digunakan. Tabel 21.4menunjukkan bahwa karakteristik lain dari unit kontrol microprogrammed juga jatuh bersamaspektrum dan bahwa spektrum ini, pada umumnya, ditentukan oleh tingkat-encodingspektrum.Sepasang barang kedua dalam tabel agak jelas. Skema Wilkes murniakan membutuhkan sebagian besar bit. Juga harus jelas bahwa ekstrem ini menyajikantampilan paling detail dari perangkat keras. Setiap sinyal kontrol dapat dikontrol secara individualoleh microprogrammer. Pengkodean dilakukan sedemikian rupa untuk mengumpulkan fungsiatau sumber daya, sehingga microprogrammer melihat prosesor pada yang lebih tinggi, kurangtingkat terperinci. Selanjutnya, pengkodean dirancang untuk memudahkan pemrograman mikrobeban. Sekali lagi, harus jelas bahwa tugas memahami dan mendalangi penggunaan semua sinyal kontrol adalah yang sulit. Seperti telah disebutkan, salah satunyakonsekuensi dari pengkodean, biasanya, adalah untuk mencegah penggunaan tertentu sebaliknyakombinasi yang diijinkan.

**Microinstruction Encoding**

Dalam prakteknya, unit kontrol yang diprogram ulang tidak dirancang menggunakan unencoded murniatau format microinstruction horizontal. Setidaknya beberapa derajat pengkodean adalah

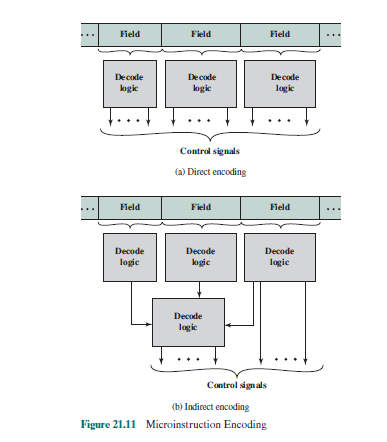
digunakan untuk mengurangi lebar memori kontrol dan untuk menyederhanakan tugasmicroprogramming. Teknik dasar untuk pengkodean diilustrasikan pada Gambar 21.11a. The microinstructiondiatur sebagai satu set bidang. Setiap bidang berisi kode, yang, di atas

decoding, mengaktifkan satu atau lebih sinyal kontrol.Mari kita pertimbangkan implikasi tata letak ini. Ketika microinstructiondijalankan, setiap bidang didekodekan dan menghasilkan sinyal kontrol. Jadi, dengan Nbidang, N tindakan simultan ditentukan. Setiap tindakan menghasilkan aktivasisatu atau lebih sinyal kontrol. Umumnya, tetapi tidak selalu, kami ingin mendesainformat sehingga setiap sinyal kontrol diaktifkan oleh tidak lebih dari satu bidang.Jelas, bagaimanapun, itu harus mungkin untuk setiap sinyal kontrol yang akan diaktifkan oleh padasetidaknya satu bidang.

Sekarang perhatikan bidang individu. Bidang yang terdiri dari bit L dapat berisi satu

kode 2L, yang masing-masing dapat dikodekan ke pola sinyal kontrol yang berbeda.

Karena hanya satu kode yang dapat muncul di suatu bidang pada satu waktu, kode-kode tersebut saling terpisah,dan, oleh karena itu, tindakan yang mereka lakukan saling eksklusif.



Desain format mikro-instruksi yang dikodekan sekarang dapat dinyatakan secara sederhana

syarat:

1. Atur format ke dalam bidang independen. Artinya, setiap bidang menggambarkan satu set

tindakan (pola sinyal kontrol) sehingga tindakan dari berbagai bidang bisa

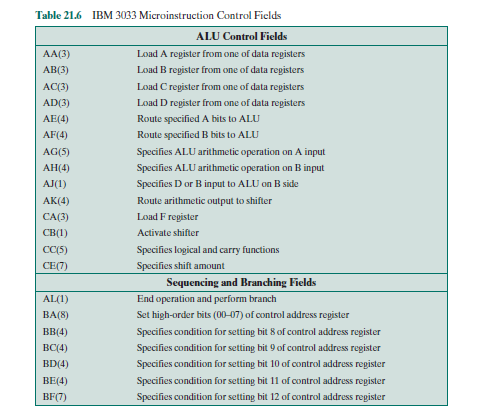
terjadi secara bersamaan.

2. Definisikan setiap bidang sedemikian rupa sehingga tindakan alternatif yang dapat ditentukan olehbidang saling eksklusif. Artinya, hanya satu dari tindakan yang ditentukan untuk a

bidang yang diberikan dapat terjadi sekaligus.

**LSI-11 Microinstruction Execution**

The LSI-11 [SEBE76] adalah contoh yang baik dari pendekatan microinstruction vertikal. Kitalihat dulu organisasi unit kontrol, lalu pada format microinstruction.organisasi unit kontrol lsi-11 LSI-11 adalah anggota pertama dariKeluarga PDP-11 yang ditawarkan sebagai prosesor single-board. Papan berisitiga chip LSI, bus internal yang dikenal sebagai bus microinstruction (MIB), dan beberapa lainnyalogika interfacing tambahan.Gambar 21.13 menggambarkan, dalam bentuk yang disederhanakan, organisasi prosesor LSI-11.Ketiga chip tersebut adalah data, kontrol, dan chip kontrol toko. Chip databerisi 8-bit ALU, dua puluh enam 8-bit register, dan penyimpanan untuk beberapa kode kondisi. Enam belas register digunakan untuk mengimplementasikan delapan 16-bit tujuan umumregister PDP-11. Lainnya termasuk kata status program, memoriregister alamat (MAR), dan register buffer memori. Karena ALU berhubungan denganhanya 8 bit pada satu waktu, dua melewati ALU diperlukan untuk menerapkanOperasi aritmatika 16-bit PDP-11. Ini dikendalikan oleh mikroprogram.



**21.4 TI 8800**

The Texas Instruments 8800 Software Development Board (SDB) adalah kartu komputer 32-bit microprogrammable. Sistem ini memiliki toko kontrol yang dapat ditulis, diimplementasikan dalam RAM daripada ROM. Sistem semacam itu tidak mencapai kecepatan atau kepadatan sistem mikroprogram dengan toko kontrol ROM. Namun demikianberguna untuk mengembangkan prototipe dan untuk tujuan pendidikan.

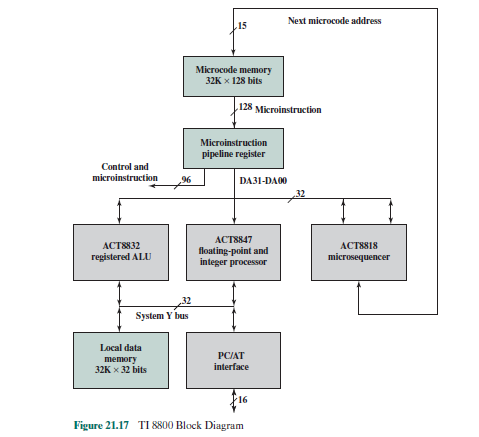
The 8800 SDB terdiri dari komponen-komponen berikut (Gambar 21.17):

* Memori mikrokode
* Microsequencer
* ALU 32-bit
* prosesor floating-point dan integer
* Memori data lokal

Dua bus menghubungkan komponen internal sistem. Bus DA menyediakandata dari bidang data microinstruction ke ALU, prosesor floating-point,atau microsequencer. Dalam kasus terakhir, data terdiri dari alamat yang akan digunakanuntuk instruksi cabang. Bus juga dapat digunakan untuk ALU atau microsequencer untuk menyediakan data ke komponen lain. Bus sistem Y menghubungkan ALU dan floatingarahkan prosesor ke memori lokal dan ke modul eksternal melalui antarmuka PC.

Papan ini cocok dengan komputer induk yang kompatibel dengan PC IBM. Komputer host

menyediakan platform yang cocok untuk perakitan dan debug mikrokoda.



**Microinstruction Format**

Format microinstruction untuk 8800 terdiri dari 128 bit yang dipecah menjadi 30

bidang fungsional, sebagaimana ditunjukkan pada Tabel 21.7. Setiap bidang terdiri dari satu atau lebih bit,

dan bidang dikelompokkan ke dalam lima kategori utama:

* Kontrol papan
* 8847 chip prosesor floating-point dan integer
* 8832 terdaftar ALU
* 8818 microsequencer
* bidang data WCS

Sebagaimana ditunjukkan pada Gambar 21.17, 32 bit bidang data WCS dimasukkan ke dalam DAbus disediakan sebagai data ke ALU, prosesor floating-point, atau microsequencer.

96 bit lainnya (field 1–27) dari microinstruction adalah sinyal kontrol yang adalangsung diumpankan ke modul yang sesuai. Untuk kesederhanaan, koneksi lain initidak ditunjukkan pada Gambar 21.17.

Enam bidang pertama berurusan dengan operasi yang berhubungan dengan kontrol dewan,daripada mengendalikan komponen individu. Operasi kontrol termasuk

berikut:

* Memilih kode kondisi untuk kontrol sequencer. Bit pertama bidang 1 menunjukkanapakah flag kondisi harus diatur ke 1 atau 0, dan 4 bit sisanyaindikasikan flag mana yang harus diatur.
* Mengirim permintaan I / O ke PC / AT.
* Mengaktifkan operasi baca / tulis memori data lokal.
* Menentukan unit yang menggerakkan bus sistem Y. Salah satu dari empat perangkatmelekat pada bus (Gambar 21.17) dipilih.

**Microsequencer**

Fungsi utama dari microsexencer 8818 adalah untuk menghasilkan microinstruction berikutnya

alamat untuk mikroprogram. Alamat 15-bit ini disediakan untuk microcodememori (Gambar21.17).

Alamat berikutnya dapat dipilih dari salah satu dari lima sumber:

1. Register mikroprogram (MPC), digunakan untuk pengulangan (digunakan kembali sama

alamat) dan terus (increment alamat oleh 1) instruksi.

2. Tumpukan, yang mendukung panggilan subroutine mikroprogram serta iterative loop dan kembali dari interupsi.

3. Port DRA dan DRB, yang menyediakan dua jalur tambahan dari eksternalperangkat keras dimana alamat mikroprogram dapat dihasilkan. Inidua port terhubung ke 16 bit paling signifikan dan paling tidak signifikanbus DA, masing-masing. Ini memungkinkan microsequencer untuk mendapatkan yang berikutnya alamat instruksi dari field data WCS dari microinstruction saat ini ataudari hasil yang dihitung oleh ALU.

4. Mendaftar penghitung RCA dan RCB, yang dapat digunakan untuk alamat tambahanpenyimpanan.

5. Input eksternal ke port Y dua arah untuk mendukung interupsi eksternal.

Stack ini memungkinkan beberapa tingkat panggilan atau interupsi bersarang, dan itu bisadigunakan untuk mendukung percabangan dan perulangan. Perlu diingat bahwa operasi ini mengacu padaunit kontrol, bukan keseluruhan prosesor, dan alamat yang terlibat adalah mereka

dari microinstructions dalam memori kontrol.Enam operasi tumpukan dimungkinkan:

1. Hapus, yang mengatur penunjuk tumpukan ke nol, mengosongkan tumpukan;

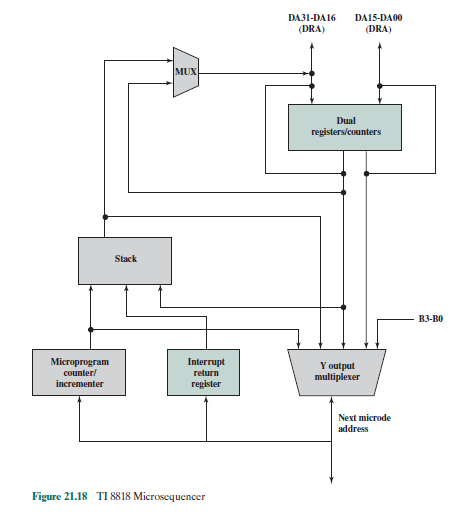
2. Pop, yang mengurangi penunjuk tumpukan;

3. Dorong, yang menempatkan isi MPC, register ulang interupsi, atau DRAbus ke tumpukan dan menambah penunjuk tumpukan;

4. Baca, yang membuat alamat yang ditunjukkan oleh pointer baca tersedia diY output multiplexer;

5. Tahan, yang menyebabkan alamat penunjuk tumpukan tetap tidak berubah;

6. Load stack pointer, yang memasukkan tujuh bit paling signifikan dari DRA kepenunjuk tumpukan.



**Registered ALU**

The 8832 adalah ALU 32-bit dengan 64 register yang dapat dikonfigurasi untuk beroperasi sebagai empatALU 8-bit, dua ALU 16-bit, atau ALU 32-bit tunggal.The 8832 dikendalikan oleh 39 bit yang membentuk bidang 17 hingga 27 darimicroinstruction (Tabel 21.7); ini dipasok ke ALU sebagai sinyal kontrol.Selain itu, seperti yang ditunjukkan pada Gambar 21.17, 8832 memiliki koneksi eksternal keBus DA 32-bit dan bus Y sistem 32-bit. Masukan dari DA dapat disediakansekaligus sebagai input data ke file register 64-kata dan ke modul logika ALU.Masukan dari bus sistem Y disediakan ke modul logika ALU. Hasil dariALU dan operasi shift adalah output ke bus DA atau bus sistem Y. Hasildapat juga diumpankan kembali ke file register internal.Tiga port alamat 6-bit memungkinkan pengambilan dua operan dan operan menulis kedilakukan dalam file register secara bersamaan. Sebuah MQ shifter dan MQ registerjuga dapat dikonfigurasi untuk berfungsi secara independen untuk menerapkan presisi ganda8-bit, 16-bit, dan 32-bit operasi shift.