SN74AS8835
16-Bit Microsequencer

- 16-bit address bus accesses up to 64K of control store
- Expandable to any multiple of 16 bits
- Compatible with 'AS890 architecture and instruction set
- 65-word stack depth
- 20-bit stack width for storage of status bits
- Stack pointer can be loaded and read externally
- Relative addressing supported by on-chip ALU that includes carry-in and carry-out for expandability
- Five groups of four status signals support multiway branching
- Internal zero status can be forced externally
- 12 external status pins with built-in pipeline register and multiplexer provide flexibility for condition code generation
- Incrementer carry-out for overflow checking and sequencer expansion
- Supports real time interrupts, trap interrupts, hardware breakpoints
- Incorporates both hardware and software resets
- Shadow registers for diagnostics
- Master/slave circuitry
- 156-pin package

Description

The 'AS8835 is a 16-bit microcontroller designed to support high-speed 'AS8832 systems. The sequencer can address up to 64K of microcode memory and is compatible with the 14-bit 'AS890. A 65-word-deep by 20-bit-wide push down stack permits address and processor status information to be stored during subroutine calls and interrupts. Diagnostic aids include shadow registers that allow the designer to read the top-of-stack, trap register, interrupt register, microprogram counter and upper and lower breakpoint registers. The stack pointer can be loaded and read from an external data port. A master/slave comparator is provided at the Y port.

Separate control of register counters, stack and source selection for the next address to be output allow the designer to merge basic operations, such as doubly nested loops, n-way branches, conditional branches and subroutine calls and returns in a large number of complex single instructions. Sixteen signals B15-B0 in combination with four external status signals provide multi-way branch capability selectable from five sources. Relative addressing schemes are supported by an ALU that adds or subtracts the contents of DRA and the microprogram counter. Eight additional external status pins and a condition code generator block provide the designer with increased flexibility for status signal and condition code select configurations.

Input to the chip is through two external data ports, DRA and DRB, or by means of the bidirectional Y port, which also outputs the current address generated by the sequencer. Other components include a 16-bit microprogram counter consisting of a register and incrementer to generate the next sequential address; two register/counters for counting loops and iterations, and storing branch addresses; a trap register; a breakpoint comparator with upper and lower limits; an interrupt return register and Y output enable for interrupt processing at the microinstruction level; and a Y output multiplexer by which the next address can be selected from the microprogram counter, register counters, external data ports DRA and DRB, relative address generator, top of stack, trap register or multi-way branch generator.

All sequencer components except the register/counters have been designed for expansion to any multiple of 16 bits. Cascading sequencer packages will increase the number of register counters by 2n, where n equals the number of cascaded packages. The 'AS8835 will be offered in a 156-pin grid array package.

Y Output Multiplexer

The Y output multiplexer of the 'AS8835 is capable of selecting the next branch address from one of ten locations. Addresses may be sourced from:

1. An external input on the DRB port, potentially a pipeline register;
2. An external input on the DRA port, potentially a pipeline register;
3. An internal microprogram counter (MPC register);
4. Internal register/counter A;
5. Internal register/counter B;
6. A multi-way branch implemented by appending four bits to DRA, DRB, register/counter A or register/counter B;
7. A branch to a relative address branch implemented by adding the contents of DRA and the MPC;
8. The top of the 65-word by 20-bit address stack;
9. Trap register;
10. An external input onto the bidirectional Y output port.

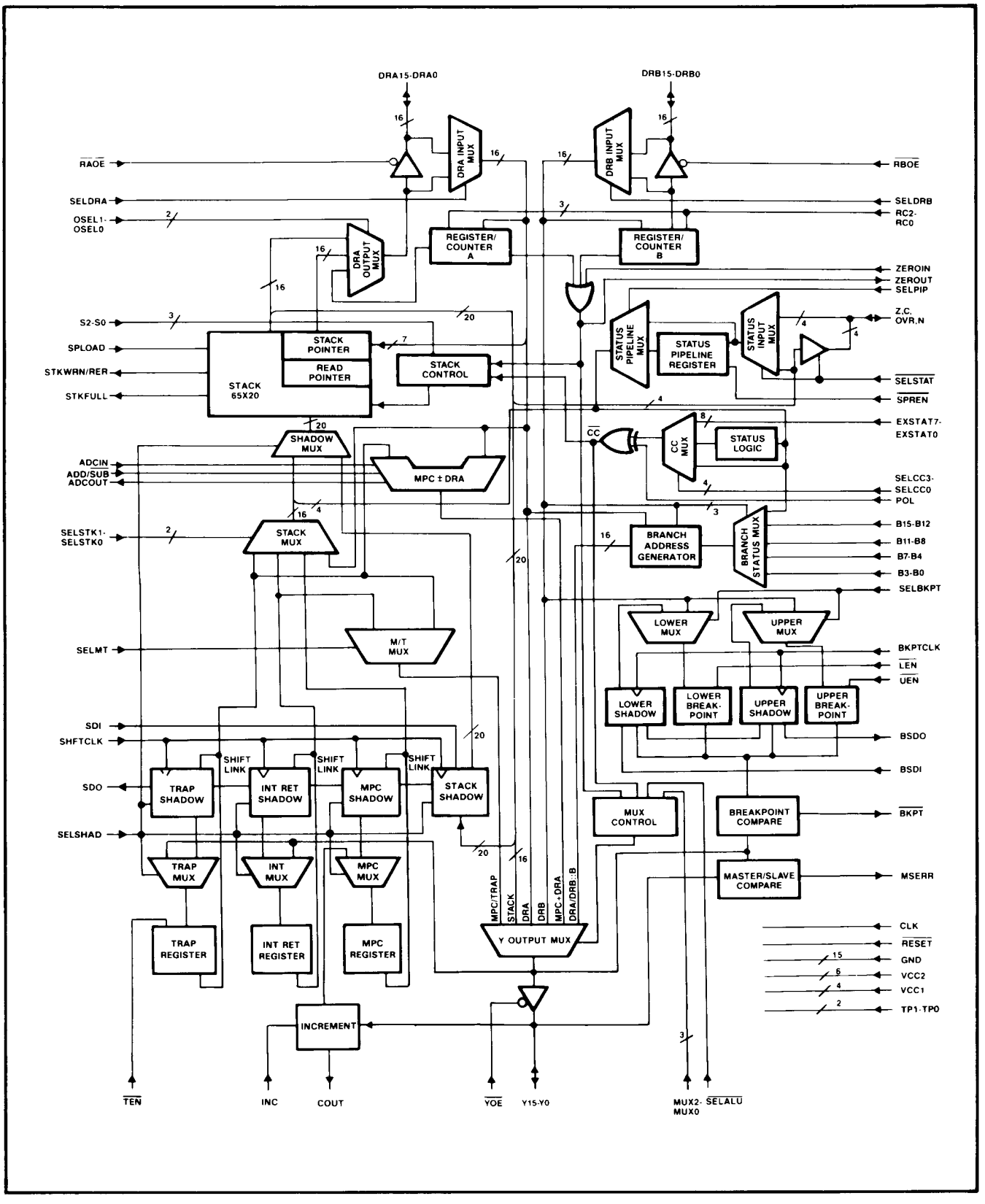

Figure 1. 74AS8835 Microsequencer

The source of the next address is dependent upon the control signals shown in Table 1. These include:

1. Stack controls S2-S0, which when low, force the Y outputs to zero;
2. $\overline{\text{SELALU}}$, which when low causes a branch to a relative address formed by adding the contents of DRA to the value in the MPC register or by subtracting DRA from MPC;
3. SELMT, which controls the MPC/Trap multiplexer;
4. ZERO, an internal status flag that indicates that one of the register/counters is being decremented to zero (This internal flag can be forced externally using the ZEROIN input);
5. Multiplexer controls (MUX2-MUX0) which select from input lines containing data from:
   a. the stack;
   b. DRA or RCA;
   c. MPC or trap;
   d. DRB or RCB;
   e. the concatenation of DRA, RCA, DRB or RCB with selected B15-B0 inputs, status from the top of stack or external inputs Z, N, C and OVR;
6. $\overline{\text{CC}}$, an internal condition code signal (see page 11).

Table 1. Y Output Control

Reset*	$\overline{\text{SELALU}}$	Multiplexer Control			Y Output Source				
					$\overline{\text{CC}}$ = L			$\overline{\text{CC}}$ = H	
					Zero = L	Zero = H			
		MUX2	MUX1	MUX0		SELMT = H	SELMT = L	SELMT = H	SELMT = L
YES	X	X	X	X	All Low	All Low	All Low	All Low	All Low
NO	L	L	L	L	STK	MPC	TRAP	MPC±DRA	MPC±DRA
NO	L	L	L	H	Reserved	Reserved	Reserved	Reserved	Reserved
NO	L	L	H	L	STK	MPC±DRA	MPC±DRA	MPC	TRAP
NO	L	L	H	H	Reserved	Reserved	Reserved	Reserved	Reserved
NO	L	H	L	L	MPC±DRA	MPC	TRAP	DRB	DRB
NO	L	H	L	H	Reserved	Reserved	Reserved	Reserved	Reserved
NO	L	H	H	L	MPC±DRA	STK	STK	MPC	TRAP
NO	L	H	H	H	Reserved	Reserved	Reserved	Reserved	Reserved
NO	H	L	L	L	STK	MPC	TRAP	DRA	DRA
NO	H	L	L	H	STK	MPC	TRAP	DRB	DRB
NO	H	L	H	L	STK	DRA	DRA	MPC	TRAP
NO	H	L	H	H	STK	DRB	DRB	MPC	TRAP
NO	H	H	L	L	DRA	MPC	TRAP	DRB	DRB
NO	H	H	L	H	DRA'**	MPC	TRAP	DRB'**	DRB'**
NO	H	H	H	L	DRA	STK	STK	MPC	TRAP
NO	H	H	H	H	DRB	STK	STK	MPC	TRAP

*A reset can be coded by pulling the S2-S0 inputs or the $\overline{\text{RESET}}$ pin low.

**DRA' and DRB' are formed by concatenating DRA15-DRA4 or DRB15-DRB4 with the four bits selected by the Branch Status multiplexer (B15-B12; B11-B8; B7-B4; B3-B0; status from the stack; or Z, N, C, OVR). The status bits are the least significant bits of the new address.

The entire instruction set can be made conditional by manipulation of the condition code (see Table 1). Allowing the condition code value to vary as a result of data or status provides for state-dependent or data-dependent branching. Unconditional branches may be achieved by causing the condition code ($\overline{CC}$) to go high (see Tables 8 and 9) and setting SELMT to select the appropriate address source. Causing $\overline{CC}$ to go low will provide for branches as dictated by the state of the zero-detect flag.

Microprogram Register and Incrementer

The microprogram register (MPC) and the incrementer (INC) provide the means for generating the next microprogram address for sequential addressing operations. The current address on the Y bus is passed to the MPC multiplexer at each rising clock edge, either unaltered (INC low) for repeating statements, or incremented by one (INC high) for addressing sequential control store locations. If SELSHAD is low, the address is passed to the MPC register; if SELSHAD is high, the address in the MPC shadow register is passed to the MPC register.

The MPC may also be externally loaded for subroutine and interrupt functions. Taking $\overline{YOE}$ high and forcing the new address onto the bidirectional Y bus loads the MPC with the new address at the positive clock edge if SELSHAD is low. This value may also be incremented prior to storage in the MPC for sequential addressing of subroutines of interrupt routines.

A carry-out on the incrementer goes high when the contents of the incrementer are FFFF and a high appears on the INC pin. On cascaded packages, the carry-out from the least significant package should be tied to INC of the next significant package.

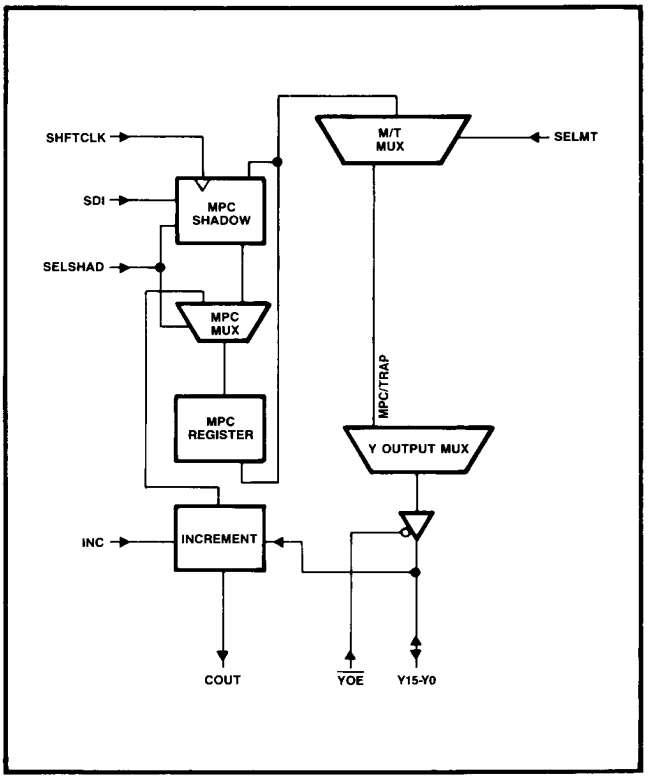

Figure 2. Microprogram Register

Register/Counters

Addresses may be loaded directly into register/counter A (RCA) and register/counter B (RCB) through the DRA and DRB ports respectively. The values stored in these registers may either be held or decremented as a result of register control inputs RC2-RC0. All combinations of these functions are supported with the exception of a simultaneous decrement of both registers (see Table 2). Generation of iteration routines may be accomplished by loading RCA and/or RCB and operating them as down counters. Loop termination is acknowledged by the ZEROUT output going high to indicate that a register contains a binary one and that a decrement is about to take

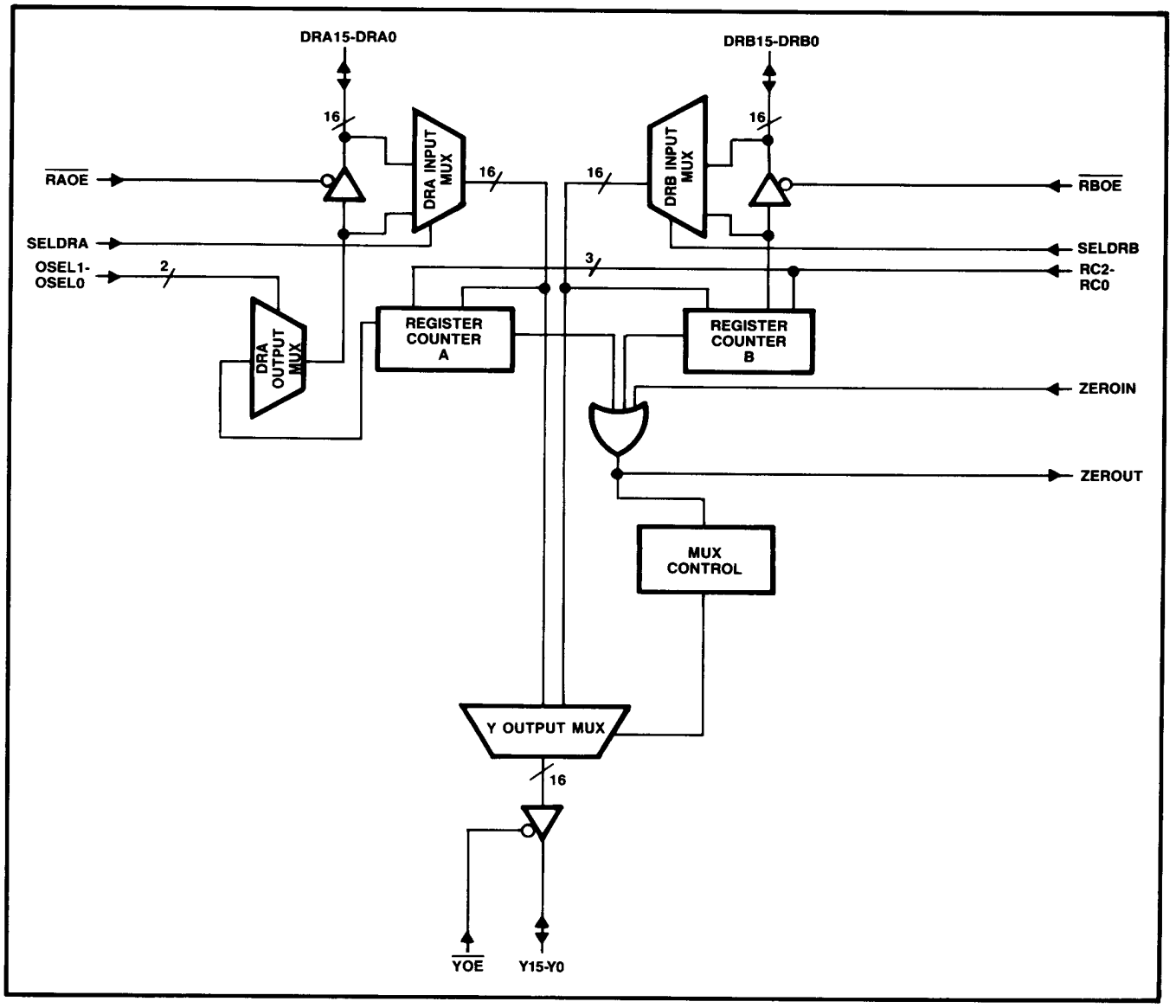

Figure 3. Register/Counters

place. Because of this facility, a "decrement and branch on zero" loop may be executed in the same clock cycle.

The contents of RCA are accessible to the DRA port when both OSEL1 and OSEL0 are low and the output bus is enabled by a low on $\overline{\text{RAOE}}$ (see Table 6). Data from RCB is available when DRB is enabled by a low on $\overline{\text{RBOE}}$. Note that the register/counters operate independently from the data buses; a value being directed from the DRA and DRB buses to the output will not inhibit the decrement operation.

Table 2. Register Control

Register Control RC2-RC0			RCA	RCB
L	L	L	Hold	Hold
L	L	H	Decrement	Hold
L	H	L	Load	Hold
L	H	H	Decrement	Load
H	L	L	Load	Load
H	L	H	Hold	Decrement
H	H	L	Hold	Load
H	H	H	Load	Decrement

Multi-Way Branch Addressing

Four groups of external inputs (B15-B12, B11-B8, B7-B4 and B3-B0) support branch addressing. The branch status multiplexer selects one of these groups for concatenation with the twelve most significant bits of DRA or DRB data from the external data ports or the register counters. In addition, four bits of data are available to the branch status multiplexer from the stack or external status inputs via the status multiplexer, providing a wide range of options for multi-way branching based on status results. The branch status multiplexer is controlled by the three least significant bits of DRB (see Table 3). This places some constraints on data being loaded into register/counter B in the same cycle that a branch from the branch status multiplexer is being executed.

Table 3. Branch Status Multiplexer Control

Signal			Data Selected
DRB2	DRB1	DRB0	
L	L	L	B15–B12
L	L	H	B11–B8
L	H	L	B7–B4
L	H	H	B3–B0
H	L	L	B15–B12 AND B11–B8
H	L	H	B15–B12 AND B7–B4
H	H	L	Status Pipeline Multiplexer
H	H	H	B7–B4 AND B3–B0

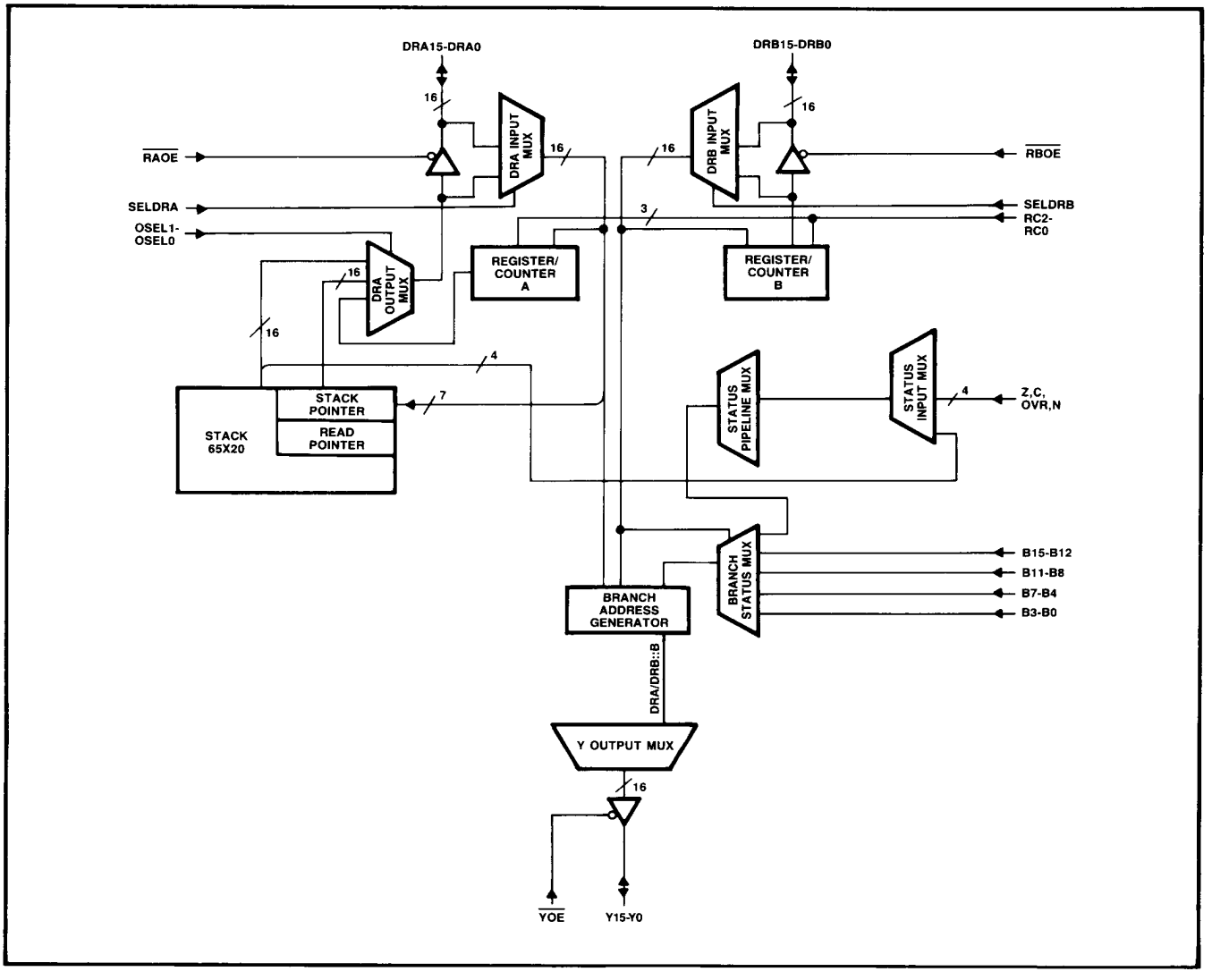

Figure 4. Branch Addressing

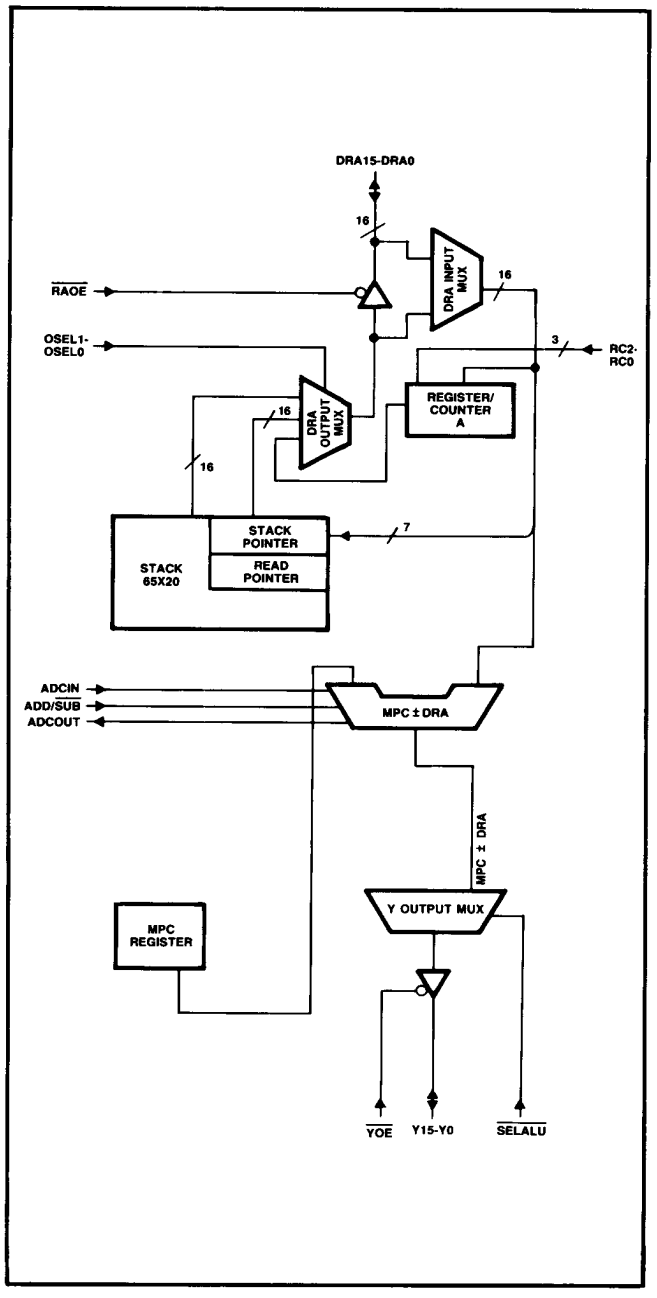

Figure 5. Relative Addressing

Relative Addressing

An ALU is provided on the 'AS8835 to facilitate relative addressing routines. The ALU adds the contents of the DRA bus to the contents of the MPC when ADD/$\overline{\text{SUB}}$ is high and subtracts DRA from MPC when ADD/$\overline{\text{SUB}}$ is low. The result is output on the Y bus when $\overline{\text{SELALU}}$ is high.

20-Bit by 65-Word Address Stack

The positive-edge-triggered address stack supplies on-chip storage of 65 control store addresses, permitting 65 nested levels of microsubroutine, looping and real-time interrupt functions. A 20-bit stack word width allows four status bits to be stored with each 16-bit address. The 16-bit address can be selected from the MPC, trap register, interrupt return register or stack shadow register or externally via the DRA port. The status bits can be selected from the status pipeline register or from status inputs Z, C, N and OVR. The stack pointer, which operates as an up-down counter, is incremented after the execution of each push operation and decremented before each pop.

In a push operation, status from the pipeline select multiplexer and data selected by SELSTK1 and SELSTK0 (see Table 4) are loaded into the stack location addressed by the stack pointer. The stack pointer is then incremented. The 16 bits from the stack multiplexer can be output at the DRA port by enabling DRA ($\overline{\text{RAOE}}$ = L, OSEL1 = L, OSEL0 = H). During a read operation (see Table 5), the four status bits are output on bidirectional status pins Z, C, OVR and N.

The stack multiplexer is controlled by two SELSTK signals and SELSHAD as shown in Table 4.

Table 4. Stack Source Controls

Signal			Data Source
SELSHAD	SELSTK1	SELSTK0	
L	L	L	MPC
L	L	H	Trap register
L	H	L	Interrupt return register
L	H	H	DRA
H	X	X	Shadow register

A pop operation causes the stack pointer to be decremented on the first rising clock edge following the arrival of the pop instruction at the S2-S0 pins. The word that was indexed by the stack pointer is effectively removed from the top of the stack. All push and pop instructions are conditionally dependent upon the stack control inputs (S2-S0), the condition code ($\overline{\text{CC}}$), and the zero-detect status. The desired option may be selected using the stack control inputs listed in Table 5.

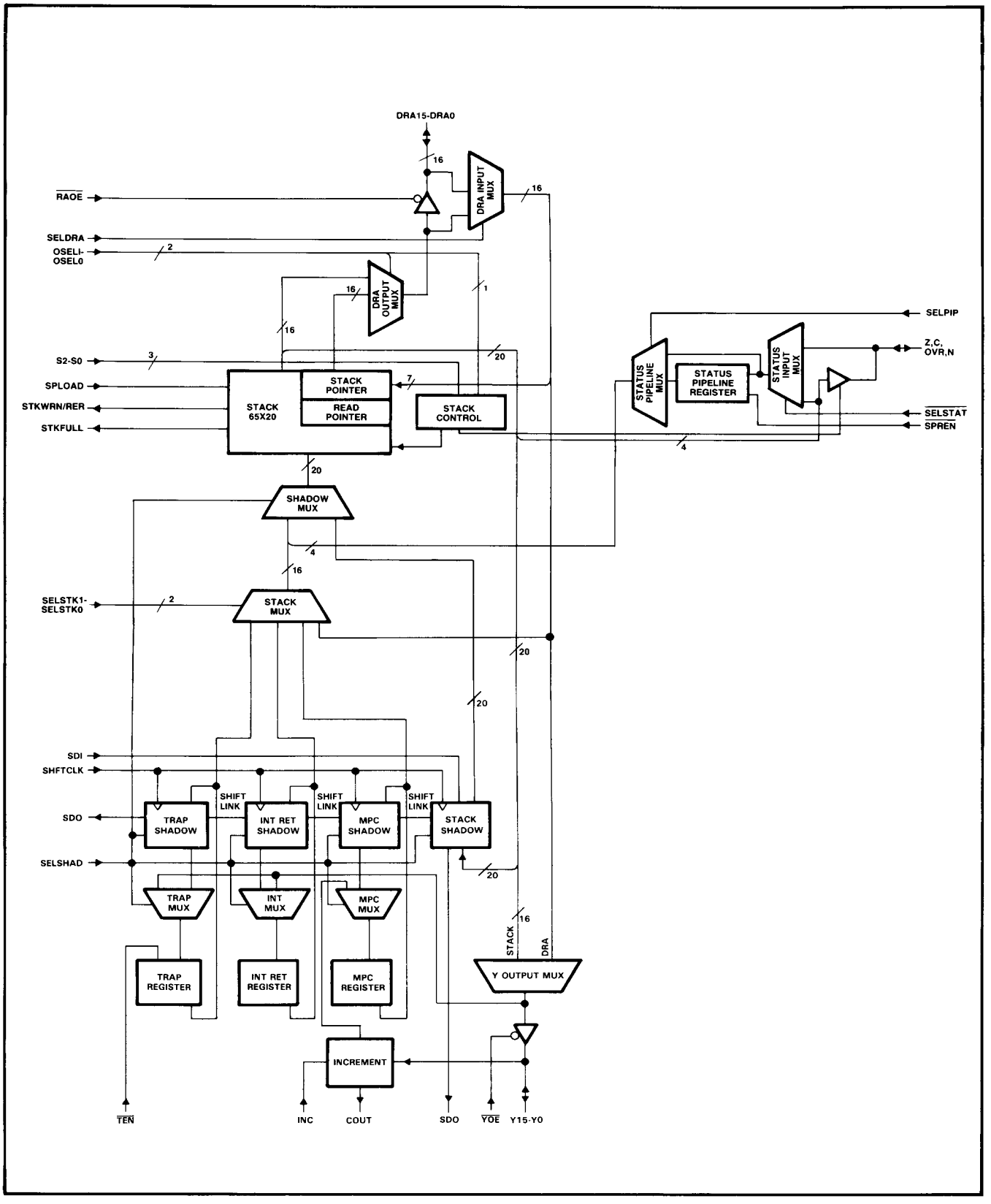

Figure 6. Stack

Stack flags

A stack warning/read error flag (STKWRN/RER) is provided to monitor the position of the stack and read pointers. During a push operation, this flag will become active when only two empty locations remain in the stack; thus reserving two locations to service asynchronous interrupts and traps. A second flag, STKFULL, will become active when the stack is full; further pushes are inhibited once the STKFULL signal goes high.

STKWRN/RER will also become active after the last address is popped from the stack, indicating that the stack is empty. During nondestructive read operations, STKWRN/RER will become active after all valid stack entries have been read.

Table 5. Stack Control

Signal		Stack Operation		
		$\overline{CC}$ = L		$\overline{CC}$ = H
S2-S0	OSEL0	Zero = L	Zero = H	
LLL	X	Reset*	Reset*	Reset*
LLH	X	Clear**	Hold	Hold
LHL	X	Hold	Pop	Pop
LHH	X	Pop	Hold	Hold
HLL	X	Hold	Push	Push
HLH	X	Push	Hold	Hold
HHL	X	Push	Hold	Push
HHH	H	Read	Read	Read
HHH	L	Hold	Hold	Hold

*Reset clears the stack and read pointers and sets the output of the Y multiplexer to zero.

**Clear performs an internal reset but does not affect the Y multiplexer output.

Stack and read pointers

The read pointer is a useful tool in debugging microcoded systems, allowing a nondestructive, sequential read of the stack contents to be performed from the DRA port. This capability provides the user with a method of backtracking through the address sequence to determine the cause of overflow without affecting program flow, the status of the stack-pointer or the internal data of the stack.

Placing a high value on all of the stack inputs (S2-S0) and setting OSEL1 low and OSEL0 high places the 'AS8835 into the read mode. At each low-to-high clock transition, the read pointer is decremented, and the 16 data bits and 4 status bits pointed to by the read pointer are available at the DRA port and the bidirectional status signals, respectively. Output from the DRA port is selected by the DRA output multiplexer, as shown in Table 6.

Table 6. DRA Output Multiplexer Controls

OSEL1	OSEL0	Output Source
L	L	Register/counter A
L	H	Stack
H	X	See Table 7

The stack pointer and stack contents are unaffected by the read operation. During push, pop and hold operations, the read pointer is updated to contain the same value as the stack pointer.

The seven least significant bits of the DRA input mux will be loaded into the stack pointer when SPLOAD is high. Loading the stack pointer externally provides options for stack expansion, split stack configurations and random stack access. During split stack operations, the STKFULL and STKWRN/RER flags may not be valid; these conditions should be monitored externally.

The values in the stack and read pointers can be read at the DRA port when OSEL1 is high (see Table 7).

Table 7. Data Available on DRA when OSEL1 = H

Signal	Value
DRA15	ZEROUT
DRA14	$\overline{CC}$
DRA13–DRA7	Read pointer
DRA6–DRA0	Stack pointer

Clearing the stack and read pointers is accomplished by placing low levels onto the stack control lines (S2-S0) or a low on the $\overline{RESET}$ pin. This function overrides all of the Y output multiplexer controls and places the Y bus into a low state. The pointers may also be cleared without affecting the Y output using the Clear instruction in Table 5.

Condition Code

Internal status logic and 12 external status pins provide a number of possible configurations for status signal and condition code select. An on-chip status pipeline register is provided for storage of status from the current stack location or external status on the zero (Z), overflow (OVR), sign (N) and carry-out (C) inputs. A status pipeline multiplexer selects the external status inputs or the pipeline register for input to a status logic block. The status logic block performs logical operations on the selected inputs (see Table 8) to provide for condition testing as shown in Table 9.

Condition code selection is controlled by four SELCC inputs (see Table 8). The selected code is exclusive ORed with the polarity signal (POL) to produce the condition code ($\overline{CC}$). The polarity signal allows the user to test input data for branch conditions (See Table 9) and invert the result if required to generate the proper $\overline{CC}$ signal for Y-output multiplexer sourcing (see Table 1).

The condition code can be read at the DRA port when OSEL1 is high (see Table 7).

Table 8. Condition Code Multiplexer Control

Signal				Signal Selected
SELCC3	SELCC2	SELCC1	SELCC0	
L	L	L	L	EXSTAT0
L	L	L	H	EXSTAT1
L	L	H	L	EXSTAT2
L	L	H	H	EXSTAT3
L	H	L	L	EXSTAT4
L	H	L	H	EXSTAT5
L	H	H	L	EXSTAT6
L	H	H	H	EXSTAT7
H	L	L	L	C*
H	L	L	H	N*
H	L	H	L	OVR*
H	L	H	H	Z*
H	H	L	L	$\overline{C}$ OR Z*
H	H	L	H	C OR Z*
H	H	H	L	N XOR OVR*
H	H	H	H	N XOR OVR OR Z*

*Output from condition code (CC) multiplexer

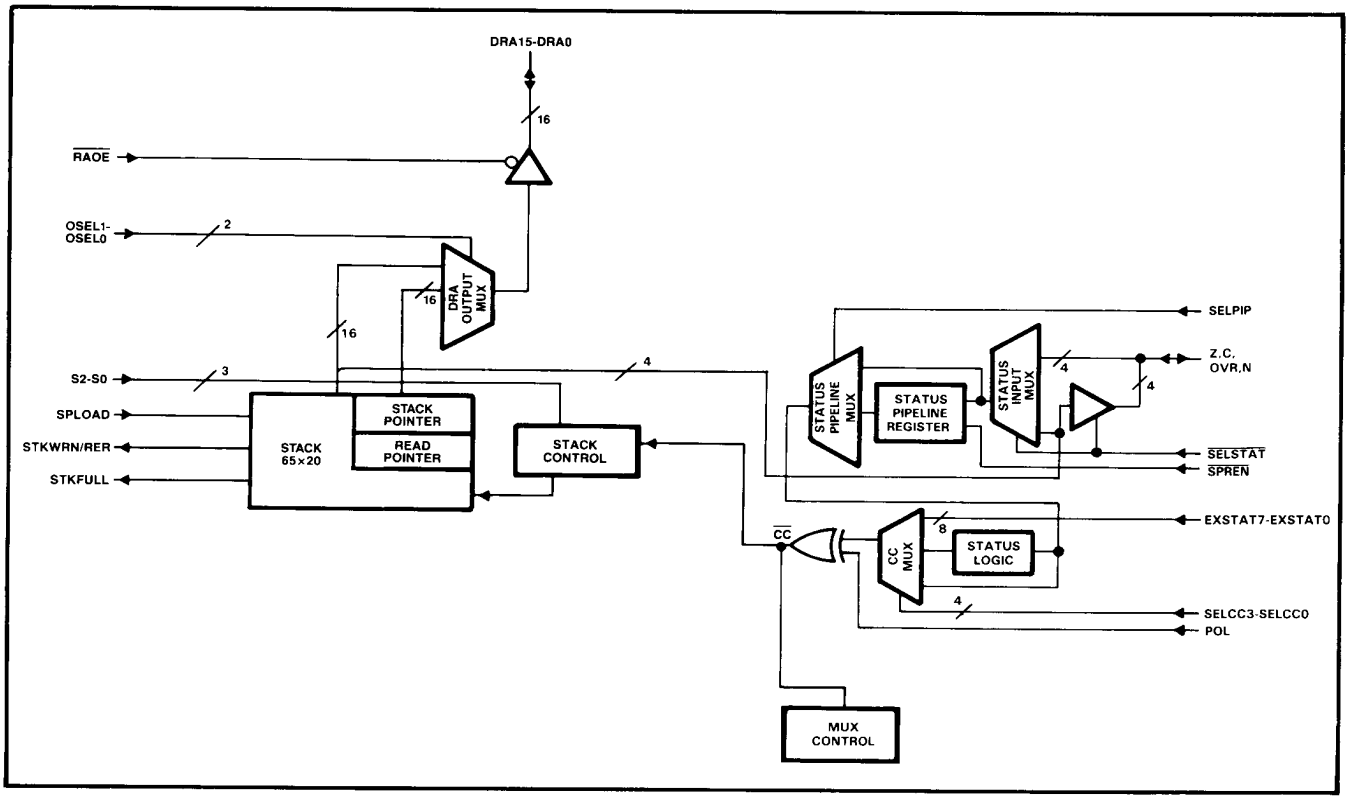

Figure 7. Condition Code

Table 9. Condition Testing Following an "A-B" Operation

Signals					Logical Operation Performed or Signal Evaluated to Generate Condition Code		Relationship Test
POL	SELCC3	SELCC2	SELCC1	SELCC0	Unsigned Numbers	2's Complement Numbers	
L	H	L	L	L	C		A≥B
L	H	L	H	H	Z	Z	A=B
L	H	H	L	L	$\overline{C}$ OR Z		A≤B
L	H	H	H	L		N XNOR OVR	A>B
L	H	H	H	H		(N XOR OVR) OR Z	A≤B
H	H	L	L	L	$\overline{C}$		A>B
H	H	L	H	H	$\overline{Z}$	$\overline{Z}$	A≠B
H	H	H	L	L	C AND $\overline{Z}$		A>B
H	H	H	H	L		N XNOR OVR	A<B
H	H	H	H	H		(N XNOR OVR) AND $\overline{Z}$	A>B

Interrupts

Real-time vectored interrupt routines are supported for those applications where polling would impede system throughput. Any instruction, including pushes and pops, may be interrupted. To process an interrupt, the following procedure should be followed:

1. The Y bus is placed into the high-impedance state by forcing $\overline{YOE}$ high.
2. The interrupt entry point vector is then forced onto the Y bus and incremented to become the second microinstruction of the interrupt routine. This is accomplished by making INC high and SELSHAD low.
3. At the following clock edge, the second microaddress is stored in the MPC and the interrupted address will be stored in the INT RET register which always contains the output from the Y multiplexer. This edge also causes the processor to begin execution of the first instruction of the interrupt routine. This first instruction must push the address stored in the INT RET register onto the stack with its corresponding status bits so that the proper return linkage is maintained and the correct condition code can be generated. This is accomplished by setting SELSHAD low, SELSTK1 high, SELSTK0 low and performing a push. If this instruction were to be interrupted, the process would be repeated and the proper return linkage preserved.

Traps

Traps may be processed using the above procedure, but using the trap register instead of the INT RET register to hold the interrupted address. Because the trap register can be disabled ($\overline{TEN}$ = H), the interrupted address does not have to be pushed onto the stack as long as nested traps are not required.

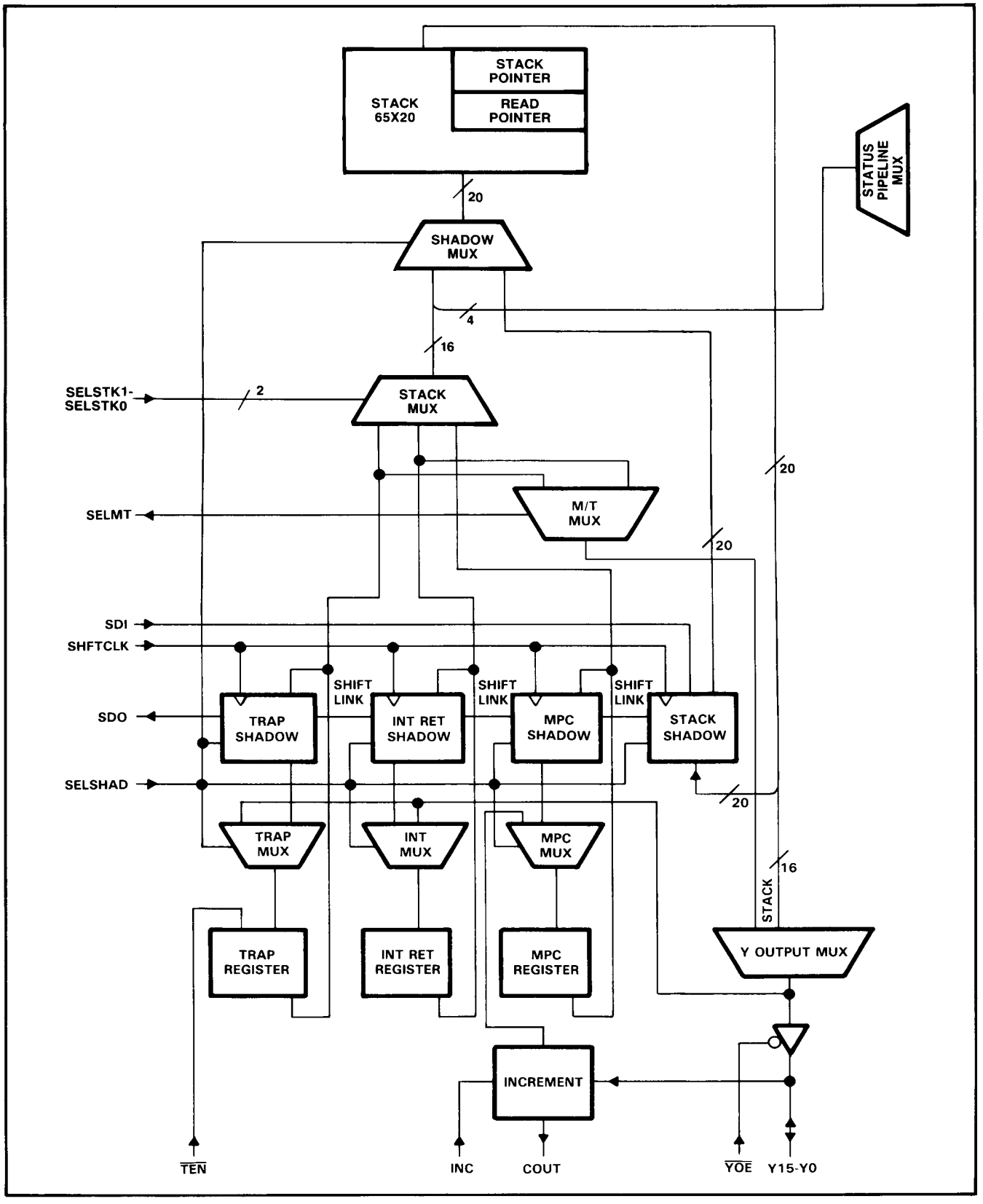

Figure 8. Interrupts and Traps

Shadow Registers

Two chains of shadow registers are provided to support system diagnostics. One set serves the breakpoint registers. The other links the stack and the MPC, trap and interrupt return registers.

Figure 10 is a generalized diagram of the registers and shadow registers, where I_n represents input data, S_n represents the shadow register output, Y_n represents register output, SDI represents the serial data input pin, SHFTCLK represents the shadow register clock and SEL represents the multiplexer control (SELSHAD or SELBKPT). Table 10 lists available load, hold and shift commands for the registers.

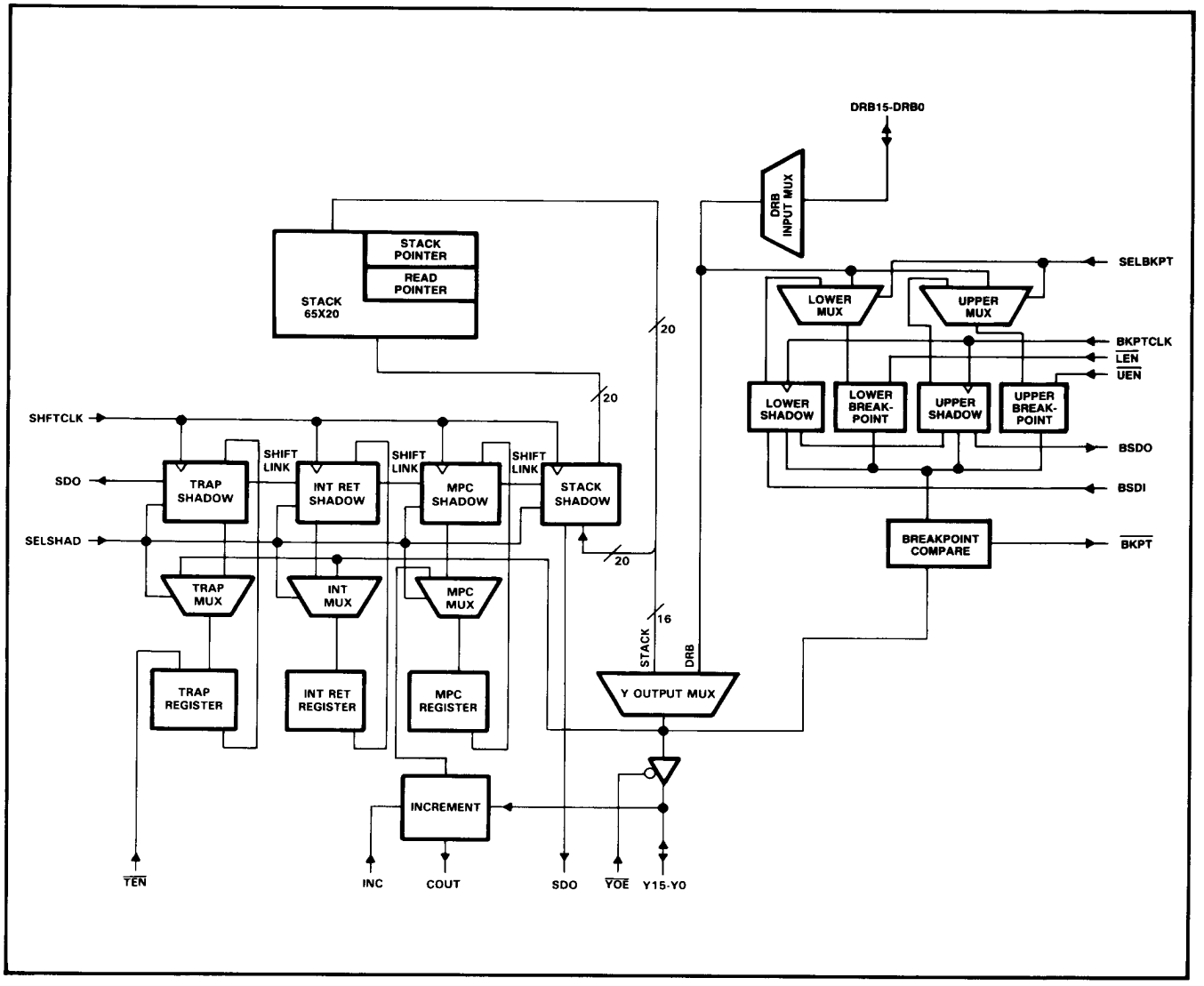

Figure 9. Shadow Registers

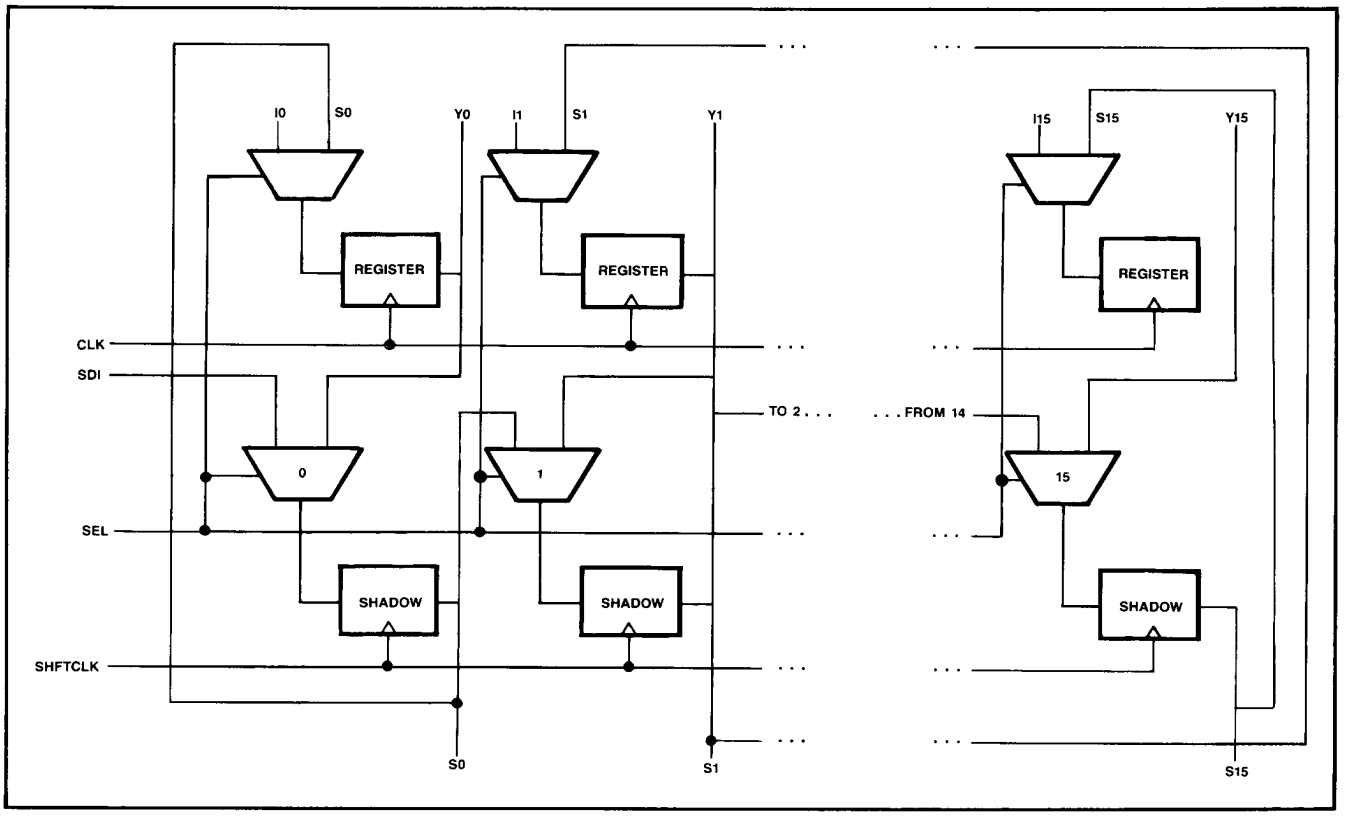

Figure 10. Generalized Register/Shadow Register Configuration

Table 10. Register/Shadow Register Controls

Inputs				Result	
SDI	SELSHAD or SELBKPT	CLK	BKPTCLK or SHFTCLK	Register	Shadow Register
X	L	⎍		Load from input bus	–
X	L		⎍	–	Serial shift right, end fill with SDI
X	L	⎍	⎍	Load from input bus	Serial shift right, end fill with SDI
L	H	⎍		Load from shadow	–
L	H		⎍	–	Load from register
L	H	⎍	⎍	Load from shadow	Load from register
H	H	⎍		Hold	–
H	H		⎍	–	Hold
H	H	⎍	⎍	Hold	Hold

NOTE: The trap and breakpoint registers have enables ($\overline{TEN}$, $\overline{LEN}$ and $\overline{UEN}$) that override the above table. If the enables are low, the registers are not altered by the above instructions.

Breakpoint Comparator

Breakpoint routines are supported by a breakpoint comparator that compares the contents of the Y output multiplexer with the values in two breakpoint registers. The registers can be used to hold the upper and lower limits of a breakpoint range. A breakpoint signal ($\overline{BKPT}$) goes low when the Y address is out of range. Tying this signal to the trap register enable ($\overline{TEN}$) provides a means of processing breakpoint routines without using the stack.

The breakpoint registers can be loaded from DRB or from the breakpoint shadow registers. External serial data can be shifted into the registers using BSDI and output on BSDO. The breakpoint shadow registers are independent of the other 'AS8835 shadow registers, and are controlled by a separate clock (BKPTCLK).

Master/Slave Comparator

A master/slave comparator is provided to support fault tolerant systems. The master/slave logic compares the contents of the Y output multiplexer with a value on the Y port. The MSERR output signal goes high when the two inputs are not equal.

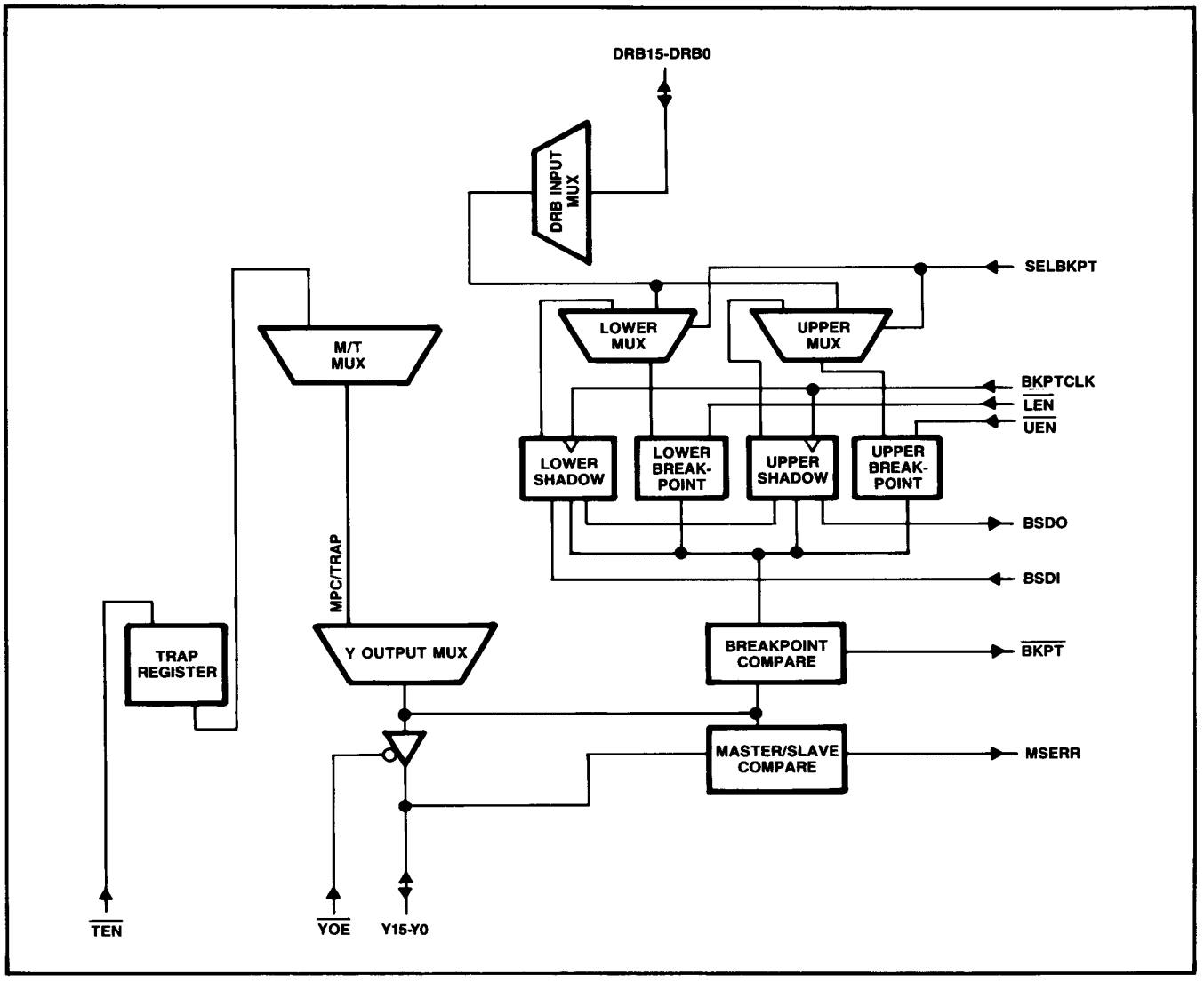

Figure 11. Breakpoint and Master/Slave Comparators

Instruction Set

Separate control of the Y-output multiplexer, stack and register/counter controls enable the user to generate a wide variety of instructions from simple sequences and branches to more complex conditional and multi-way branches, subroutine jumps and calls. The required signals for selection of address sources were listed in Table 1. Controls for the register/counters and stack operations were summarized in Tables 3 and 5. Table 11 summarizes the required signals for control of 'AS8835 multiplexers and data inputs and outputs. These tables have been combined to implement a few commonly used microsequencing instructions as shown in Table 12.

Table 11. Response to Control Inputs

Name	Logic Level	
	High	Low
INC	Adds one to Y output	Passes Y output to MPC multiplexer
$\overline{\text{LEN}}$	Disables lower breakpoint register load	Enables lower breakpoint register load
MUX2-MUX0	See Table 1	See Table 1
OSEL1-OSEL0	See Table 6	See Table 6
POL	Inverts condition code multiplexer output	Passes condition code multiplexer output
$\overline{\text{RAOE}}$	Places DRA output in high-Z state	Enables DRA output
$\overline{\text{RBOE}}$	Places DRB output in high-Z state	Enables DRB output
RC2-R0	See Table 2	See Table 2
$\overline{\text{SELALU}}$	Selects Y-output source from MUX2-MUX0	Selects relative address (DRA ± MPC)
SELBKPT	Selects breakpoint shadow register	Selects DRB
SELCC3-SELCC0	See Table 8	See Table 8
SELDRA	Selects external DRA port	Selects DRA output multiplexer
SELDRB	Selects register counter B	Selects external DRB port
SELMT	Selects MPC	Selects trap register
SELPIP	Selects status pipeline register	Selects status multiplexer
SELSHAD	Selects shadow registers for MPC, trap, interrupt, stack	Selects MPC, trap, interrupt, stack registers
SELSTAT	Selects external status Z, C, OVR, N	Selects stack
SELSTK1-SELSTK0	See Table 4	See Table 4
SPLOAD	Selects externally-generated stack pointer value (7 LSBs of DRA)	Selects stack pointer value generated by push, pop or hold operation
$\overline{\text{SPREN}}$	Disables status pipeline register input	Enables status pipeline register input
S2-S0	See Table 5	See Table 5
$\overline{\text{TEN}}$	Disables trap register load	Enables trap register load
$\overline{\text{UEN}}$	Disables upper breakpoint register load	Enables upper breakpoint register load
$\overline{\text{YOE}}$	Disables Y output bus	Enables Y output bus
ZEROIN	Forces internal zero detect signal high	Enables internal zero detect signal

Table 12. Control Signals for Common Sequencing Instructions

Instruction	$\overline{SELALU}$	SELMT	MUX2-MUX0	S2-S0	R2-R0	$\overline{CC}$*	INC
Clear Pointers	1	X	X X X	0 0 1	X X X	0	X
Continue	1	1	1 1 0	1 1 1	X X X	1	1
Continue and Pop	1	1	1 1 0	0 1 0	X X X	1	1
Continue and Push	1	1	1 1 0	1 0 0	X X X	1	1
Branch to DRA	1	X	0 0 0	1 1 1	X X X	1	X
Multi-way Branch on DRB'**	1	X	1 0 1	1 1 1	X X X	1	X
Multi-way Branch on DRA' else DRB'**	1	X	1 0 1	1 1 1	0 0 0	X	X
Conditional Branch (Branch to DRA else Continue)	1	1	1 1 0	1 1 1	0 0 0	V	1
Three-way Branch (Decrement RCA; Branch to DRA else Branch to DRB else Continue)	1	1	1 0 0	1 1 1	0 0 1	V	1
Repeat using MPC	1	1	1 1 0	1 1 1	X X X	1	0
Repeat using Stack	1	X	0 1 0	1 1 1	0 0 0	0	X
Repeat until $\overline{CC}$=H (Branch to Stack else Continue)	1	1	0 1 0	1 1 1	0 0 0	V	1
Loop until Zero (Decrement RCA; Branch to Stack else Continue and Pop)	1	1	0 0 0	0 1 0	0 0 1	0	1
Conditional Loop until Zero (Decrement RCA; Branch to Stack else Continue else Branch to DRA and Pop)	1	1	0 0 0	0 1 0	0 0 1	V	1
Jump to Subroutine (Branch to DRA and Push)	1	X	0 0 0	1 1 0	X X X	1	1
Conditional Jump to Subroutine (Branch to DRA and Push else Continue)	1	1	1 1 0	1 0 1	0 0 0	V	1
2-Way Jump to Subroutine (Branch to DRA and Push else Branch to DRB and Push)	1	X	1 0 0	1 1 0	0 0 0	V	1
Return from Subroutine (Branch to Stack and Pop)	1	X	0 1 0	0 1 1	0 0 0	0	X
Conditional Return from Subroutine (Branch to Stack and Pop else Continue)	1	1	0 1 0	0 1 1	0 0 0	V	1
Reset	X	X	X X X	0 0 0	X X X	X	X

*$\overline{CC}$ is an internal condition code. To force $\overline{CC}$ high or low, see Table 8. An "X" in this column indicates a don't care condition. A "V" means that $\overline{CC}$ is variable, for example originating from system status results rather than from microcode.

**DRA' and DRB' are branch addresses formed by the twelve most significant bits of concatenating the 12 MSBs of DRA or DRB with branch status inputs (see Table 3).

Specifications

Preliminary specifications for the 'AS8835 are given below. Table 13 shows absolute maximum ratings. Preliminary performance data are given in Table 14. Tables 15 and 16 contain recommended operating conditions and electrical characteristics.

Table 13. Absolute Maximum Ratings Over Operating Temperature Range (unless otherwise noted)

Supply voltage, VCC1	7 V
Supply voltage, VCC2	3 V
Input voltage: All inputs I/O ports	7 V 5.5 V
Operating free-air temperature range	0°C to 70°C
Storage temperature range	−65°C to 150°C

Table 14. 'AS8835 Preliminary Performance Data

Data Path	Typical Timing (NS)
DRA/DRB → Y	11
MUX2-MUX0 → Y	12
R2-R0 → Y	15
R2-R0 → ZEROUT	15
S2-S0 → Y	9
MPC → Y	12
MPC±DRA → Y	25
TEST → Y	16
CLK → Y	20

Table 15. Recommended Operating Conditions

Parameter			SN74AS835			Unit
			MIN	NOM	MAX	
V_{CC1}	I/O supply voltage		4.5	5	5.5	V
V_{CC2}	STL internal logic supply voltage		1.8	2	2.2	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				08	V
I_{OH}	High-level output current				−2.6	mA
I_{OL}	Low-level output current	Flag and status outputs			8	mA
		Y, DRA, DRB			24	
T_A	Operating free air temperature		0		70	°C

Table 16. Electrical Characteristics Over Recommended Operating Temperature Range (unless otherwise noted)

Parameter		Test Conditions		SN74AS835			Unit
				MIN	TYP†	MAX	
V_{IK}		V_{CC1} = 4.5 V,	I_I = 18 mA			−1.2	V
V_{OH}		V_{CC1} = 4.5 V to 5.5 V,	I_{OH} = 0.4 mA	VCC−2			V
		V_{CC1} = 4.5 V,	I_{OH} = −2.6 mA	2.4			
V_{OL}	Flag, status	V_{CC1} = 4.5 V,	I_{OL} = 8 mA			0.5	V
	Y, DRA, DRB	V_{CC1} = 4.5 V,	I_{OL} = 12 mA			0.5	
I_I	I/O ports	V_{CC1} = 5.5 V,	V_I = 5.5 V			0.1	mA
	All other	V_{CC1} = 5.5 V,	V_I = 7 V			0.1	
I_{IH}	I/O ports‡	V_{CC1} = 5.5 V,	V_I = 2.7 V			40	μA
	All other					20	
I_{IL}	I/O ports, ‡	V_{CC1} = 5.5 V,	V_I = 0.4 V			−0.4	mA
	All other					−0.4	
I_{O}§		V_{CC1} = 5.5 V,	V_O = 2.25 V	−30		−112	mA
I_{CC1}		V_{CC1} = 5.5 V			250		mA
I_{CC2}		V_{CC2} = 2.1 V			780		mA

†All typical values are at V_{CC} = 5 V. T_A = 25°C.

‡For I/O ports the parameters I_{IH} and I_{IL} include output current I_{OZH} and I_{OZL} respectively.

§The output conditions have been chosen to produce a current that closely approximates one-half of the true short circuit I_{OS}.