Table 3-10. R-Bus Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| NOP (No Operation) | 1111 | No Operation. |
| MREG | 0011 | The MREG R-bus field code is used to fetch a memory element that happens to lie in a TOS register (i.e., E is greater than SM). Prior to executing MREG, the value S minus E must be placed in the SP1 register. During execution of MREG, TNAME becomes the sum of NAME and SP1(14:15) and the R-bus register is loaded as follows:<br>If TNAME = 00 then R-BUS := TR0R<br>If TNAME = 01 then R-BUS := TR1R<br>If TNAME = 10 then R-BUS := TR2R<br>If TNAME = 11 then R-BUS := TR3R<br>Due to the pipeline affect, a TOS register referenced in the store field of the preceding microinstruction assumes the above TNAME. |
| PADD (Pre-Adder) | 0100 | The 16-bit output of the pre-adder is loaded into the R-bus register. |
| PL (Program Limit) | 0000 | The 16-bit content of the program limit (PL) register is loaded into the R-bus register. |
| RA | 1011 | The RA R-bus field code is used to read the content of the first TOS register (location S). SR must be greater than 0.* During execution, TNAME becomes NAME and the R-bus register is loaded as follows:<br>If TNAME = 00 then R-BUS := TR0R<br>If TNAME = 01 then R-BUS := TR1R<br>If TNAME = 10 then R-BUS := TR2R<br>If TNAME = 11 then R-BUS := TR3R |
| RB | 1010 | The RB R-bus field code is used to read the second TOS register (location S-1). SR must be greater than 1.* During execution, TNAME becomes NAME and the R-bus register is loaded as follows:<br>If TNAME = 00 then R-BUS := TR1R<br>If TNAME = 01 then R-BUS := TR2R<br>If TNAME = 10 then R-BUS := TR3R<br>If TNAME = 11 then R-BUS := TR0R |
| RBUS | 0101 | The RBUS R-bus field code causes the R-bus register to remain unchanged. |
| RC | 1001 | The RC R-bus field code is used to read the third TOS register (location S2). SR must be greater than 2.* During execution, TNAME becomes NAME and the R-bus register is loaded as follows:<br>If TNAME = 00 then R-BUS := TR2R<br>If TNAME = 01 then R-BUS := TR3R<br>If TNAME = 10 then R-BUS := TR0R<br>If TNAME = 11 then R-BUS := TR1R |

*True only if RA:RD are being used as part of the stack. RA:RD often are used by the microprogram as scratch pad registers when not used otherwise.

Table 3-10. R-Bus Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| RD | 1000 | The RD R-bus field code is used to read the fourth TOS register (location S-3). SR must be equal to 4.* During execution, TNAME becomes NAME and the R-bus register is loaded as follows:<br><br>If TNAME = 00 then R-BUS := TR3R<br>If TNAME = 01 then R-BUS := TR0R<br>If TNAME = 10 then R-BUS := TR1R<br>If TNAME = 11 then R-BUS := TR2R |
| SP0<br>(Scratch Pad 0) | 1101 | The 16-bit content of the scratch pad 0 (SP0) register is loaded into the R-bus register. |
| SP1<br>(Scratch Pad 1) | 1100 | The 16-bit content of the scratch pad 1 (SP1) register is loaded into the R-bus register. |
| SR<br>(Stack Register) | 0001 | The 3-bit content of the stack (SR) register is loaded into the R-bus register bits 0 thru 2. R-bus register bits 3 thru 15 become zeros. |
| UBUS | 1110 | The 16-bit U-bus data word is loaded into the R-bus register. The U-bus data is established by the preceding microinstruction. |
| X<br>(Index) | 0110 | The 16-bit content of the index (X) register is loaded into the R-bus register. |
| XC<br>(X Conditional) | 0111 | The XC R-bus field code is used with indexed memory addressing. If the index bit of the current instruction (CIR bit 4) is zero, the R-bus register is loaded with zeros, otherwise the R-bus register is loaded with the 16-bit content of the X-register. |
| Z<br>(Stack Limit) | 0010 | The 16-bit content of the stack limit (Z) register is loaded into the R-bus register. |

*True only if RA:RD are being used as part of the stack. RA:RD often are used by the microprogram as scratch pad registers when not used otherwise.

Table 3-3. S-Bus Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| (blank) | 11111 | The S-bus register is loaded with all zeros. |
| CC (Condition Code) | 10111 | The CC S-bus field code is used to retrieve the condition code (CC) portion of the status word for use with certain conditional branch instructions. When executed, bits 6 and 7 of the status word are loaded into bits 8 and 9 of the S-bus register and if both of these bits are zeros, S-bus register bit 7 becomes a one. All other S-bus register bits become zeros. |
| CIR (Current Instruction Register) | 00000 | The 16-bit output of the current instruction register (CIR) is loaded into the S-bus register. |
| CPX1 | 00100 | CPX1, a collection of 16 special signals, is loaded into the S-bus register. |
| CPX2 | 00110 | CPX2, a collection of 16 special signals, is loaded into the S-bus register. |
| CTRH (Counter High) | 01101 | The 6-bit content of the counter (CNTR) register is loaded into bits 4 thru 9 of the S-bus register. All other S-bus register bits become zeros. |
| CTRL (Counter Low) | 01100 | The 6-bit content of the counter (CNTR) register is loaded into bits 10 thru 15 of the S-bus register. All other S-bus register bits become zeros. |
| DB (Data Base) | 10101 | The 16-bit content of the data base (DB) register is loaded into the S-bus register. |
| DL (Data Limit) | 11100 | The 16-bit content of the data limit (DL) register is loaded into the S-bus register. |
| IOA (I/O Address) | 01001 | The 8-bit content of the interrupt device number (IDN) register is loaded into bits 8 thru 15 of the S-bus register. Bits 0 thru 7 of the S-bus register become zeros. |
| IOD (I/O Data) | 01010 | The 16-bit content of the direct input data (DID/MUXMA) register in the IOP is loaded into the S-bus register. |
| MOD (Module Number) | 00101 | The MOD S-bus field code provides the CPU with two pieces of information. When executed, the 4-bit content of the interrupt module number (IMN) register is loaded into bits 4 thru 7 of the S-bus register. Also, if the CPU is CPU1, bit 13 of the S-bus register becomes a 1. If CPU2 (Series II only), bit 12 of the S-bus register becomes a 1. These bits are used to fetch the correct Q1 and Z1 entries in the code segment table. All other bits of the S-bus register become zeros. |
| NOP (No Operation) | 11111 | No operation. |
| OPND (Operand) | 10110 | The 16-bit content of the operand (OPND) register is loaded into the S-bus register. An attempt to execute an OPND while an MCU operand directed operation is in progress results in a CPU freeze until the MCU operation is complete. |

Table 3-3. S-Bus Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| P (Program Counter) | 10000 | The 16-bit content of the program counter (P) register is loaded into the S-bus register. |
| PADD (Pre-Adder) | 00010 | The 16-bit output of the pre-adder (PADD) is loaded into the S-bus register. |
| PB (Program Base) | 11110 | The 16-bit content of the program base (PB) register is loaded into the S-bus register. |
| PCLK (Process Clock) | 01011 | The Process Clock, PCLK, is placed in the S-bus register. |
| Q (Stack Marker Pointer) | 10001 | The 16-bit content of the stack marker pointer (Q) register is loaded into the S-bus register. |
| QDWN (Stack Marker Pointer Down) | 01000 | The QDWN S-bus field code is used to put the content of the lowest valid TOS register in the S-bus register. During execution, TNAME becomes the sum of NAME and SR(1:2) and the S-bus register is loaded as follows:<br><br>If TNAME = 00 then S-BUS := TR3S<br>If TNAME = 01 then S-BUS := TR0S<br>If TNAME = 10 then S-BUS := TR1S<br>If TNAME = 11 then S-BUS := TR2S<br><br>To preserve stack integrity, a DCSR (Decrement SR) code is normally executed. Due to the pipeline effect, a TOS reference in the store field of the preceding microinstruction also uses the above described TNAME. |
| RA | 11011 | The RA S-bus field code is used to read the content of the first TOS register (location S). SR must be greater than zero.* During execution, TNAME becomes NAME and the S-bus register is loaded as follows:<br><br>If TNAME = 00 then S-BUS := TR0S<br>If TNAME = 01 then S-BUS := TR1S<br>If TNAME = 10 then S-BUS := TR2S<br>If TNAME = 11 then S-BUS := TR3S |
| RB | 11010 | The RB S-bus field code is used to read the content of the second TOS register (location S-1). SR must be greater than 1.* During execution, TNAME becomes NAME and the S-bus register is loaded as follows:<br><br>If TNAME = 00 then S-BUS := TR1S<br>If TNAME = 01 then S-BUS := TR2S<br>If TNAME = 10 then S-BUS := TR3S<br>If TNAME = 11 then S-BUS := TR0S |
| RBR (Read Bank Register) | 00011 | Read bank register onto S-bus (14:15 for Series II and 12:15 for Series III). The S-bus bits 0 - 13 are zeroed. The bank register to be read is specified in the MCU field. Execution of the Special field is inhibited. |

Table 3-3. S-Bus Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| RC | 11001 | The RC S-bus field code is used to read the content of the third TOS register (location S-2). SR must be greater than 2.* During execution, TNAME becomes NAME and the S-bus register is loaded as follows:<br><br>If TNAME = 00 then S-BUS := TR2S<br>If TNAME = 01 then S-BUS := TR3S<br>If TNAME = 10 then S-BUS := TR0S<br>If TNAME = 11 then S-BUS := TR1S |
| RD | 11000 | The RD S-bus field code is used to read the content of the fourth TOS register (location S-3). SR must be equal to 4.* During execution, TNAME becomes NAME and the S-bus register is loaded as follows:<br><br>If TNAME = 00 then S-BUS := TR3S<br>If TNAME = 01 then S-BUS := TR0S<br>If TNAME = 10 then S-BUS := TR1S<br>If TNAME = 11 then S-BUS := TR2S |
| SBUS | 01111 | The SBUS code causes the S-bus register content to remain unchanged. |
| SM (Stack Memory) | 10011 | The 16-bit content of the stack memory (SM) register is loaded into the S-bus register. |
| SP1 (Scratch Pad 1) | 00001 | The 16-bit content of the scratch pad 1 (SP1) register is loaded into the S-bus register. |
| SP2 (Scratch Pad 2) | 11101 | The 16-bit content of the scratch pad 2 (SP2) register is loaded into the S-bus register. |
| SP3 (Scratch Pad 3) | 10101 | The 16-bit content of the scratch pad 3 (SP3) register is loaded into the S-bus register. |
| STA (Status) | 10100 | The 16-bit status word is loaded into the S-bus register. |
| SWCH | 00111 | The 16-bit content of the switch register is loaded into the S-bus register. |
| UBUS | 01110 | The 16-bit U-bus data word is loaded into the S-bus register. The U-bus data is established by the preceding microinstruction. |

*True only if RA:RD are being used as part of the stack. RA:RD often are used by the microprogram as scratch pad registers when not used otherwise.

Table 3-5. Function Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| ADD | 11111 | The content of the R-bus register is added to the content of the S-bus register and the result is placed on the T-bus. |
| ADDO (Add-Enable Overflow) | 11011 | The content of the R-bus register is added to the content of the S-bus register and the result is placed on the T-bus. Carry and overflow are modified in the status register and CCA is set on the T-bus. |
| AND | 00111 | The content of the R-bus register is logically "anded" with the content of the S-bus register and the result is placed on the T-bus. |
| BNDT (Bounds Test) | 01101 | The function field code BNDT is used to perform a bounds test of an address. Execution of this code results in the content of the R-bus register minus the content of the S-bus register being placed on the T-bus. If RRZ, RLZ, LRZ, LLZ is specified, then BNDT does a "CAD" instead of a "SUB." The R- and S-bus fields are coded so that this result is a negative number (CARRY = 0) if a bounds violation occurs. If the CPU is not operating in the priviledged mode (STATUS(0) = 0), and a bounds violation occurs, a microjump to ROM address 0003 is executed. If no violation has occurred (CARRY = 1) or the CPU is operating in the priviledged mode (STATUS(0) = 1), the next microinstruction will be executed in the usual manner. |
| CAD (Complement and Add) | 01110 | The content of the R-bus register is added to the one's complement of the content of the S-bus register and the result is placed on the T-bus. If the S-bus register contains all zeros, CAD results in the R-bus register content minus 1 on the T-bus. |
| CADO (Complement and Add-Enable Overflow) | 01010 | The content of the R-bus register is added to the one's complement of the content of the S-bus register and the result is placed on the T-bus. Carry and overflow are modified in the status register and the condition code is set to CCA on the T-bus data. |
| CAND (Complement-And) | 00101 | The R-bus register content is logically "anded" with the complement of the S-bus register content and the result is placed on the T-bus. |
| CRS (Circular Shift) | 11010 | The R-bus register content is added to the S-bus register content and the result is placed on the T-bus. The T-bus is then circular shifted one place right or left as specified in the shift field (SR1 or SL1) and placed on the U-bus. |
| CTSD (Controlled Shift Double) | 10111 | The function field code CTSD adds the contents of the R-bus register and the S-bus register, puts the result on the T-bus, and performs a double word shift of the T-bus and a scratch pad left or right as specified by the shift field code (SR1 or SL1). The type of shift is determined by the content of the current instruction register (CIR) as follows:<br><br>If CIR(7) = 1 then circular shift<br>If CIR(7:8) = 01 then logical shift<br>If CIR(7:8) = 00 then arithmetic shift<br><br>The most significant word is on the T-bus. If a left shift is specified, scratch pad 1 (SP1) contains the least significant word. If a right shift is specified, scratch pad 3 (SP3) contains the least significant word. Regardless of the direction of the shift, both SP1 and SP3 are shifted left and right respectively. |

Table 3-5. Function Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| CTSS (Controlled T-bus Shift Single) | 11100 | The R-bus register content is added to the S-bus register content and the result is placed on the T-bus. The T-bus is then shifted left or right as specified by the shift field code (SR1 or SL1). The type of shift is determined by the content of the current instruction register as follows:<br><br>If CIR(7) = 1 then circular shift<br>If CIR(7:8) = 01 then logical shift<br>If CIR(7:8) = 00 then arithmetic shift |
| DCAD (Decimal Add) | 11110 | The contents of the R- and S-bus registers are added and the results are placed into the decimal correction adder. The decimal corrector adder output is placed on the U-bus. |
| DVSB (Divide-Subtract) | 01000 | The function field code DVSB performs the subtract, shift, and test necessary to execute a divide algorithm. The R- and S-bus fields of the microinstruction are coded so that initially the 16-bit divisor is in the S-bus register and the most significant 16-bits of the dividend are in the R-bus register. The least significant 16-bits of the dividend are in the SP1 register. Both divisor and dividend must be positive numbers upon execution of the DVSB code and Flag 2 (F2) must be 0 (cleared). An SL1 code in the shift field of the microinstruction directs the left shift of the T-bus. The follcwing algorithm is then executed repeatedly to perform the complete divide.<br><br>TBUS := RBUS - SBUS;<br>UBUS(0:14) := TBUS(1:15);<br>If ALU carry or F2=1 then<br>BEGIN<br>RREG(0:14) := UBUS(0:14);<br>RREG(15) := SP1(0);<br>SP1(0:14) := SP1(1:15);<br>SP1(15) := 1<br>F2 := TBUS(0);<br>END<br>else<br>BEGIN<br>RREG(0:14) := RREG(1:15);<br>RREG(15) := SP1(0);<br>SP1(0:14) := SP1(1:15);<br>SP1(15) := 0<br>F2 := RREG(0);<br>END<br><br>For example, after 17 executions of the above algorithm, a 16-bit quotient is contained in the SP1 register and the remainder times 2 is contained in the R-bus register. When the remainder is unloaded from the R-bus register, it is shifted right one place (divided by 2). |

Table 3-5. Function Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| INC (Incremented Add) | 11101 | The R-bus register content is added to the S-bus register content plus 1. The result is placed on the T-bus. |
| INCO (Incremented Add-Enable Overflow) | 11001 | The R-bus register content is added to the S-bus register content plus 1, the result placed on the T-bus, and the carry and overflow is modified in the status register. The condition code is set to CCA on the T-bus data. |
| IOR (Inclusive OR) | 10110 | The content of the R-bus register is logically inclusively "ored" with the content of the S-bus register and the result is placed on the T-bus. |
| JMP (Jump) | 01100 | The JMP function field code directs a micro-jump to the ROM address (jump target) specified by bits 20 thru 31 of the ROM output register if the skip field condition is met (a condition must be specified). The R-bus, shift, and special field decoders are disabled and the U-bus and T-bus become the S-bus register content. |
| JSB (Jump to Subroutine) | 00100 | The JSB function field code directs a micro-subroutine jump to the ROM address specified by bits 20 thru 31 of the ROM output register if the skip field code condition is met. If the condition is met and the JSB is executed, the save register is loaded with the address of the line following the JSB and is used as a return address at the subroutine end (see function field code RSB). During execution of the JSB, the R-bus, shift, and special field decoders are disabled and the T-bus and U-bus become the S-bus register content. |
| MPAD (Multiply-Add) | 11000 | The function field code MPAD performs the add, shift, and test necessary to execute a multiply algorithm. The R-bus field of the microinstruction is coded so that initially the 16-bit multiplicand is in the R-bus register. The S-bus field code is UBUS which is initially all zeros. Scratch pad 3 contains the 16-bit multiplier. Both multiplier and multiplicand must be positive numbers upon execution of the MPAD code. An SR1 code in the shift field directs the right shift of the T-bus. The following algorithm is executed repeatedly to perform a complete multiply.<br><br>T-BUS := R-REG plus S-REG;<br>U-BUS(1:15) := T-BUS(0:14);<br>U-BUS(0) := ALUcarry;<br>If SP3(15) = 1 then<br>BEGIN<br>S-REG := U-BUS;<br>SP3(1:15) := SP3(0:14);<br>SP3(0) := T-BUS(15);<br>END<br>else |

Table 3-5. Function Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| | | BEGIN<br>S-REG(1:15) := S-REG(0:14);<br>SP3(1:15) := SP3(0:14);<br>SP3(0) := S-REG(15);<br>END<br><br>After 16-executions of the above algorithm, the result is a 32-bit word with the most significant 16-bits in the S-bus register and the least significant 16-bits in scratch pad 3. |
| PNLR<br>(Panel Read) | 10000 | The PNLR function field code allows the auxiliary control panel to select and display a CPU register. This code appears in the microprogram during execution of HALT and PAUSE routines. When PNLR is executed, the ROM output register (ROR1) R-bus and S-bus fields are disabled. The maintenance panel interface supplies these field codes which put the content of the selected register in the associated (R- or S-bus) register. The T-bus and U-bus become the R-bus register content plus the S-bus register content (one of which will be zeros). The auxiliary control panel completes this operation by displaying the U-bus as the selected register. |
| PNLS<br>(Panel Store) | 10001 | The PNLS function field code allows the auxiliary control panel to load a CPU register with the content of one of its switch registers. This code is part of the halt mode interrupt micro-routine for servicing a maintenance panel interrupt. When PNLS is executed, the ROM output register (ROR2) store field is disabled and the maintenance panel interface card supplies the store field code respective of the selected CPU register. A SWCH S-bus field code causes the S-bus register to be loaded with the content of the selected auxiliary control panel switch register. The T-bus and U-bus become the R-bus register content (zeros) plus the S-bus register content and at the end of the cycle, the selected register is loaded with the U-bus data. |
| QASL | 00000 | The QASL function field code causes a four register arithmetic shift left of the U-bus, scratch pad 3, scratch pad 1, and the R-bus register containing the most, next most, next least, and least significant word respectively. Shift Left One code (SL1) is required in the shift field. The sign bit is preserved.<br><br>T-BUS := S-REG;<br>U-BUS(0) := T-BUS(0);<br>U-BUS(1:14) := T-BUS(2:15);<br>U-BUS(15) := SP3(0);<br>SP3(0:14) := SP3(1:15);<br>SP3(15) := SP1(0);<br>SP1(0:14) := SP1(1:15);<br>SP1(15) := R-REG(0);<br>R-REG(0:14) := R-REG(1:15);<br>R-REG(15) := 0 |

Table 3-5. Function Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| QASR | 00001 | The QASR function field code causes a four register arithmetic shift right of the U-bus, scratch pad 3, scratch pad 1, and the S-bus register containing the most, next most, next least, and least significant words respectively. Shift right one code (SR1) is required in the shift field. The sign bit is propagated.<br><br>T-BUS := R-REG;<br>U-BUS(0:1) := T-BUS(0);<br>U-BUS(2:15) := T-BUS(1:14);<br>SP3(0) := T-BUS(15);<br>SP3(1:15) := SP3(0:14);<br>SP1(0) := SP3(15);<br>SP1(1:15) := SP1(0:14);<br>S-REG(0) := SP1(15);<br>S-REG(1:15) := S-REG(0:14); |
| REPC<br>(Repeat Until Condition) | 10100 | The REPC function field code causes the next microinstruction to be executed repeatedly until the skip field condition of that microinstruction is met. During execution the T-bus becomes the R-bus register content plus the S-bus register content. The REPC code is decoded from ROR2 and at that time disables the RAR increment function and the ROR1 load function and sets the Repeat FF. The RAR then contains the address of the microinstruction following the one to be repeated and ROR1 contains the microinstruction to be repeated. The next cycle loads ROR2 and executes the microinstruction to be repeated. As long as the Repeat FF remains set, the content of ROR1 and ROR2 does not change and is executed each cycle. When the skip field condition is met, the Repeat FF is cleared, the pipeline is filled correctly, and the next microinstruction is fetched in the usual manner. |
| REPN<br>(Repeat N Times) | 10101 | The REPN function code operates in the same manner as the REPC code described for the preceding label. The difference is that REPN loads a repeat counter register with the content of the microinstruction skip field. Bits 1 thru 5 of the counter become ROR2 bits 5 thru 19; bit 0 of the counter becomes a 1. The counter content is then the two's complement of the number of repeats to be performed. To utilize the counter, the repeated microinstruction contains a special field code INCTR (Increment Counter) and a skip field condition CTRM (Counter Maximum). |
| ROM | 10011 | The function field code ROM loads the R-bus register with a 16-bit constant obtained from the microinstruction. The ROM code is decoded from ROR1, loading the R-bus register with bits 16 thru 31 of ROR1. The T-bus then becomes the R-bus register content plus the S-bus register content. The R-bus, shift, special, and skip field decoders are disabled by the ROM code. |
| ROMI<br>(ROM Inclusive) | 10010 | The function field code ROMI loads the R-bus register with a 16-bit constant obtained from the microinstruction. The ROMI code is decoded from ROR1, loading the R-bus register with ROR1 bits 16 thru 31. The T-bus then becomes the R-bus register content inclusive "ored" with the S-bus register content. The R-bus, shift, special, and skip field decoders are disabled by the ROMI code. |

Table 3-5. Function Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| ROMN (ROM And) | 00011 | The function field code ROMN loads the R-bus register with a 16-bit constant obtained from the microinstruction. The ROMN code is decoded from ROR1, loading the R-bus register with ROR1 bits 16 thru 31. The T-bus becomes the R-bus register content logically "anded" with the S-bus register content. The R-bus, shift, special, and skip field decoders are disabled by the ROMN code. |
| ROMX (ROM Exclusive) | 00010 | The function field code ROMX loads the R-bus register with a 16-bit constant obtained from the microinstruction. The ROMX code is decoded from ROR1, loading the R-bus register with ROR1 bits 16 thru 31. The T-bus becomes the R-bus register content exclusive "ored" with the S-bus register content. The R-bus, special, shift, and skip field decoders are disabled by the ROMX code. |
| SUB (Subtract) | 01111 | The content of the S-bus register is subtracted from the content of the R-bus register and the result is placed on the T-bus. |
| SUBO (Subtract-Enable Overflow) | 01011 | The content of the S-bus register is subtracted from the content of the R-bus register and the result is placed on the T-bus. Carry and overflow are modified in the status register and condition code CCA is set on the T-bus data. |
| UBNT (Unconditional Bounds Test) | 01001 | The function field code UBNT is used to perform an unconditional bounds test of an address. Execution of this code results in the content of the R-bus register minus the content of the S-bus register being placed on the T-bus. If RRZ, RLZ, LRZ, LLZ is specified, then UBNT does a "CAD" instead of a "SUB." The R-bus and S-bus field are coded so that this result is a negative number (CARRY = 0) if a bounds violation occurs. The response to a bounds violation is a micro-jump to ROM address 0003. If no violation occurs (CARRY = 1), the next microinstruction is executed in the usual manner. |
| XOR (Exclusive OR) | 00110 | The content of the R-bus register is exclusive "ored" with the content of the S-bus register and the result is placed on the T-bus. |

**Table 3-7.** Shift Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| (blank) | 111 | The T-bus word is placed directly on the U-bus. |
| LLZ (Left to Left and Zero) | 001 | The shift field code LLZ places the left byte of the T-bus in the left byte of the U-bus and zeros in the right byte of the U-bus. |
| LRZ (Left to Right and Zero) | 000 | The shift field code LRZ places the left byte of the T-bus in the right byte of the U-bus and zeros in the left byte of the U-bus. |
| RLZ (Right to Left and Zero) | 101 | The shift field code RLZ places the right byte of the T-bus in the left byte of the U-bus and zeros in the right byte of the U-bus. |
| SWAB (Swap Bytes) | 110 | The shift field code SWAB places the left byte of the T-bus in the right byte of the U-bus and the right byte of the T-bus in the left byte of the U-bus. |
| RRZ (Right to Right and Zero) | 100 | The shift field code RRZ places the right byte of the T-bus in the right byte of the U-bus and zeros in the left byte of the U-bus. |
| SL1 (Shift Left 1) | 010 | The shift field code SL1 shifts the T-bus one place left onto the U-bus. Refer to the function field code descriptions for the action taken when used with function field codes CRS, CTSD, CTSS, DVSB, and TASL. |
| SR1 (Shift Right 1) | 011 | The shift field code SR1 shifts the T-bus logically one place right onto the U-bus. Refer to the function field code descriptions for the action taken when used with function field codes CRS, CTSD, CTSS, MPAD, and TASR. |

**Table 3-4.** Store Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| NOP (No Operation) | 11111 | No Operation. |
| BSP0 (Bus to Scratch Pad Ø) | 00101 | The store field code BSP0 stores the U-bus into ACOR or DCOR, depending on the MCU field option selected and in SP0. Disables the special field and enables the MCU options, one of which must be used. |
| BSP1 (Bus to Scratch Pad 1) | 00100 | Same as BSP0 except SP1 is used. |
| BUS | 00111 | Same as BSP0 except none of the scratch-pad registers are used. |
| CTRH (Counter High) | 01111 | The store field code CTRH stores U-bus bits 4 thru 9 in the counter (CNTR) register bits 0 thru 5. |
| CTRL (Counter Low) | 01110 | The store field code CTRL stores U-bus bits 10 thru 15 in the counter (CNTR) register bits 0 thru 5. |
| DB (Data Base) | 10011 | The store field code DB stores the 16-bit U-bus word in the data base (DB) register. |
| DL (Data Limit) | 11100 | The store field code DL stores the 16-bit U-bus word in the data limit (DL) register. |
| IOA (I/O Address) | 00001 | The store field code IOA sends the command on the U-bus, bits 5 through 7, to the device whose address is on the U-bus in bits 8 through 15. U-bus, bit 0 = 1, sends a service-out signal to the device. |
| IOD (I/O Data) | 00010 | The store field code IOD stores the 16-bit word currently on the U-bus in the direct output data (DOD) register. |
| MREG (Memory Register) | 00011 | The store field code MREG is used to store data in an address that lies in a TOS register (i.e., S ≥ E > SM where S = SR + SM). Prior to executing MREG, the value E minus S is placed in the SP1 register. During execution, TNAME becomes the sum of NAME and SP1 (14:15) and the TOS registers are loaded as follows:<br>If TNAME = 00 then TR0 := U-BUS<br>If TNAME = 01 then TR1 := U-BUS<br>If TNAME = 10 then TR2 := U-BUS<br>If TNAME = 11 then TR3 := U-BUS<br>Due to the pipeline effect, a TOS register referenced in the R- or S-bus field of the following microinstruction assumes the above described TNAME. |
| P (Program Count) | 10000 | The store field code P stores the 16-bit U-bus word in the program counter (P) register. |
| PB (Program Base) | 11110 | The store field code PB stores the 16-bit U-bus word in the program base (PB) register. |

Table 3-4. Store Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| PCLK (Process Clock) | 00000 | The Process Clock, PCLK, is placed in the S-bus register. |
| PL (Program Limit) | 01001 | The store field code PL stores the 16-bit U-bus word in the program limit (PL) register. |
| PUSH | 01000 | The store field code PUSH effectively moves all stack elements down one location and loads the U-bus word on the top of stack. To maintain stack integrity, SR must be less than four. When PUSH is executed, TNAME becomes NAME and the TOS registers are loaded as follows:<br>If TNAME = 00 then TR3 := U-BUS<br>If TNAME = 01 then TR0 := U-BUS<br>If TNAME = 10 then TR1 := U-BUS<br>If TNAME = 11 then TR2 := U-BUS<br>To complete the operation, NAME is decremented and SR is incremented. |
| Q (Stack Marker Pointer) | 10001 | The store field code Q stores the 16-bit U-bus word in the stack marker pointer (Q) register. |
| QUP (Stack Marker Pointer Up) | 01011 | The store field code QUP effectively inserts the U-bus word into the stack at location SM plus one. For example, if stack locations S and S minus one are in the TOS registers and location S minus two is the first stack element in memory (location SM), execution of QUP places the U-bus word in a TOS register at stack location S minus two. The first stack element in memory (location SM) becomes S minus three. To maintain stack integrity, the SR register must be incremented (special field code INSR) indicating the addition of a TOS register element. Normally, SR should be less than four.<br>When the store field code QUP is executed, TNAME becomes the sum of NAME and SR and the TOS registers are loaded as follows:<br>If TNAME = 00 then TR0 := U-BUS<br>If TNAME = 01 then TR1 := U-BUS<br>If TNAME = 10 then TR2 := U-BUS<br>If TNAME = 11 then TR3 := U-BUS<br>Due to the pipeline effect, a TOS register referenced in the R- or S-bus fields of the following microinstruction assumes the above described TNAME. |

Table 3-4. Store Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| RA | 11011 | The store field code RA stores the U-bus word in the first TOS register (location S). SR must be greater than zero.* During execution of RA, TNAME becomes NAME and the TOS registers are loaded as follows:<br><br>If TNAME = 00 then TR0 := U-BUS<br>If TNAME = 01 then TR1 := U-BUS<br>If TNAME = 10 then TR2 := U-BUS<br>If TNAME = 11 then TR3 := U-BUS |
| RAR (ROM Address Register) | 10111 | The store field code RAR stores bits 0 thru 15 of the U-bus in ROM address register bits 0 thru 15. The intent of this code is to force the processor to a new microprogram address specified by the U-bus word. Execution of the RAR code requires three microcycles. The first loads the ROM address register and the next two are NOPs allowing the ROM output registers (ROR1 and ROR2) to be loaded with the new microinstruction. |
| RB | 11010 | The store field code RB stores the U-bus word in the second TOS register (location S minus one). SR must be greater than one.* During execution of RB, TNAME becomes NAME and the TOS registers are loaded as follows:<br><br>If TNAME = 00 then TR1 := U-BUS<br>If TNAME = 01 then TR2 := U-BUS<br>If TNAME = 10 then TR3 := U-BUS<br>If TNAME = 11 then TR0 := U-BUS |
| RC | 11001 | The store field code RC stores the U-bus word in the third TOS register (location S minus two). SR must be greater than two.* During execution of RC, TNAME becomes NAME and the TOS registers are loaded as follows:<br><br>If TNAME = 00 then TR2 := U-BUS<br>If TNAME = 01 then TR3 := U-BUS<br>If TNAME = 10 then TR0 := U-BUS<br>If TNAME = 11 then TR1 := U-BUS |
| RD | 11000 | The store field code RD stores the U-bus word in the fourth TOS register (location S minus three). SR must be equal to four.* During execution of RD, TNAME becomes NAME and the TOS registers are loaded as follows:<br><br>If TNAME = 00 then TR3 := U-BUS<br>If TNAME = 01 then TR0 := U-BUS<br>If TNAME = 10 then TR1 := U-BUS<br>If TNAME = 11 then TR2 := U-BUS |
| SBR (Stack Bank Register) | 00110 | The store field code SBR stores the U-bus bits (14:15 for Series II and 12:15 for Series III) in the bank register specified in the MCU field code. Execution of the special field is inhibited. |

*True only if RA:RD are being used as part of the stack, RA:RD often are used by the microprogram as scratch pad registers when not used otherwise.

Table 3-4. Store Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| SM (Stack Memory Pointer) | 10010 | The store field code SM stores the U-bus word in the stack memory (SM) register. |
| SP0 (Scratch Pad 0) | 01101 | The store field code SP0 stores the U-bus word in the scratch pad 0 (SP0) register. |
| SP1 (Scratch Pad 1) | 01100 | The store field code SP1 stores the U-bus word in the scratch pad 1 (SP1) register. |
| SP2 (Scratch Pad 2) | 11101 | The store field code SP2 stores the U-bus word in the scratch pad 2 (SP2) register. |
| SP3 (Scratch Pad 3) | 10101 | The store field code SP3 stores the U-bus word in the scratch pad 3 (SP3) register. |
| STA (Status) | 10100 | The store field code STA stores the U-bus word in the status register. |
| X (Index) | 10110 | The store field code X stores the U-bus word in the index (X) register. |
| Z (Stack Limit Pointer) | 01010 | The store field code Z stores the U-bus word in the stack limit pointer (Z) register. |

Table 3-8. Special Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| (blank) | 11111 | No special field operation. |
| CCA (Condition Code A) | 11110 | The special field code CCA sets the condition code bits to CCL (01) if the T-bus word is less than zero (T(0) = 1), CCE (10) if the T-bus word is equal to zero (Signal T = 0 is true), or CCG (00) if the T-bus word is greater than zero (T(0) = 0 and signal T = 0 is false). |
| CCB (Condition Code B) | 00000 | The special field code CCB sets the condition code to CCL (01) if bits 8 thru 15 of the U-bus form a special ASCII character, CCE (10) if an alphabetic ASCII character, or CCG (00) if a numeric ASCII character. |
| CCE (Condition Code E) | 11101 | The special field code CCE sets the condition code bits to CCE (10). |
| CCG (Condition Code G) | 11100 | The special field code CCG sets the condition code bits to CCG (00). |
| CCL (Condition Code L) | 11011 | The special field code CCL sets the condition code bits to CCL (01). |
| CCPX (Clear CPX1) | 00001 | Clears the interrupt status register bits as specified by the true bits on the U-bus.<br>U-Bus bit 0 Halt<br>1 Run<br>2 System Halt<br>3 (Unused)<br>Bits 4 (MSB) through 7 (LSB) code the following functions:<br>Octal Code 0 NOP<br>1 Clear BNDV<br>2 Clear Illegal Address<br>3 Clear CPU Timer<br>4 Clear System Parity Error<br>5 Clear Address Parity Error<br>6 Clear Data Parity Error<br>7 Clear Module Interrupt<br>10 Clear External Interrupt<br>11 Power Fail Turn-Off Interrupt<br>16 Reverse System Parity<br>17 Reverse MCUD Parity<br>8 Diagnostic NIRTOCIR<br>9 (Unused)<br>10 Diagnostic Set CPX1 (Bits 1:8)<br>11 Clear ICS Flag<br>12 Clear DISP Flag |

Table 3-8. Special Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| | | 13 (Unused)<br>14 Diagnostic Freeze<br>15 Clear Panel FF's |
| CCRY (Clear Carry) | 10101 | The special field code CCRY clears carry in the status register. |
| CCZ (Condition Code Zero) | 11010 | The special field code CCZ sets the condition code bits to CCE (10) if the T-bus word is equal to zero (signal T = 0 true) or CCG (00) if the T-bus word is not equal to zero (signal T = 0 false). |
| CF1 (Clear Flag 1) | 10010 | The special field code CF1 clears CPU Flag 1 FF. |
| CF2 (Clear Flag 2) | 10001 | The special field code CF2 clears CPU Flag 2 FF. |
| CF3 (Clear Flag 3) | 00111 | The special field code CF3 clears CPU Flag 3 FF. |
| CLIB (Clear Indirect Bit) | 01110 | At the end of the cycle, CLIB sets the Indirect Bit FF which masks the indirect line until a NEXT or JLUI option in the skip field is encountered. |
| CLO (Clear Overflow) | 11001 | The special field code CLO clears the status word overflow bit. |
| CLSR (Clear SR) | 00010 | The special field code CLSR clears the SR register. This is an asynchronous reset. No other SR operation is allowed during that time. |
| CTF (Set Carry to Flag 1) | 00110 | The special field code CTF stores the ALU carry bit in the Flag 1 FF. |
| DCSR (Decrement SR) | 01001 | The special field code DCSR decrements the content of the SR register by a count of one. |
| FHB (Flag to High Bit) | 01101 | The special field code FHB transfers the content of the Flag 1 FF to bit 0 of the U-bus. |
| HBF (High Bit to Flag 1) | 01100 | The special field code HBF transfers the content of U-bus bit 0 to the Flag 1 FF. |
| INCN (Increment Name) | 01010 | The special field code INCN increments the content of the name register by a count of one. |
| INCT (Increment Counter) | 01011 | The special field code INCT increments the content of the counter register by a count of one. |
| INSR (Increment SR) | 01000 | The special field code INSR increments the content of the SR register by a count of one. |
| LBF (Low Bit to Flag 2) | 01111 | The special field code LBF transfers the content of U-bus bit 15 to the Flag 2 FF. |

Table 3-8. Special Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| NOP (No Operation) | 10111 | No operation. |
| POP | 10111 | The special field code POP moves the stack elements up one location such that the second element of the stack (S minus one) becomes the top element (S), etc. The previous top of stack element is lost. When executed, this is accomplished by decrementing the SR register and incrementing the name register. |
| POPA (Pop setting CCA) | 10110 | The special field code POPA functions the same as special field code POP with the addition that the condition code is set to CCL (01) if the T-bus word is less than zero, CCE (10) if the T-bus word is equal to zero, or CCG (00) if the T-bus word is greater than zero. |
| SCRY (Set Carry Bit) | 10100 | The special field code SCRY sets Carry in the status register. |
| SDFG (Set Dispatcher Flag) | 00101 | The special field code SDFG sets the dispatcher flag (bit 12 of interrupt status register CPX1). |
| SF1 (Set Flag 1) | 10011 | The special field code SF1 sets CPU Flag 1 FF. |
| SF2 (Set Flag 2) | 10000 | The special field code SF2 sets CPU Flag 2 FF. |
| SF3 (Set Flag 3) | 00011 | The special field code SF3 sets CPU Flag 3 FF. |
| SIFG (Set Interrupt Stack Flag) | 00100 | The special field code SIFG sets the interrupt flag (bit 11 of interrupt status register CPX1). |
| SOV (Set Overflow) | 11000 | The special field code SOV sets the status word overflow bit. |

Table 3-9. MCU Option Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| ABS (Absolute) | 00000 | The MCU option code ABS specifies the Absolute bank register which may be read into S-bus (14:15 for Series II and 12:15 for Series III) with "RBR" or stored from U-bus (14:15 for Series II and 12:15 for Series III) with "SBR". This bank register is used as a scratch pad bank register by the micro code. |
| CMD (Command) | 00010 | The MCU option code CMD enables the bus options (BUS, BSP0, and BSP1) in the store field to store the U-bus into the address CPU output register, ACOR, and to initiate a low-request command. When "selected", the ACOR is output to the MCU-bus, and the command and module number ("TO" lines) are obtained from the CRL register. |
| CRL (Control) | 00001 | The MCU option code CRL enables the store field bus options (BUS, BSP0, and BSP1) to load the S-bit CRL register from the U-bus as follows:<br>CRL(0:1) := U-BUS(10:11) Command<br>CRL(2:4) := U-BUS(13:15) Address.<br>The CPU freezes until any pending MCU requests are completed. |
| DATA | 10001 | The MCU option code DATA enables the store field bus options (as in CRL) to store the U-bus into DCOR, and initiates a "high-request" command. |
| DB (DB - Relative) | 10000 | The MCU option code DB functions the same as ABS except that it specifies the DB-bank register used with DB - relative addressing. |
| DPOP (Data - POP) | 10010 | The DPOP MCU option code functions the same as DATA and also pops the stack. |
| NIR (Next Instruction Register) | 01001 | The MCU option code NIR enables the store field bus options (as in CRL) to store the U-bus into DCOR, and initiates a "high-request" command. On the following "select" cycle, DCOR is read into the MCU bus and stored in the CPU NIR register. |
| OPND (Operand Register) | 11001 | Same as NIR except that the MCU bus is stored in the CPU OPND register. |
| PB (PB - relative addressing) | 01000 | Same as ABS, except that it specifies the PB-bank register used in PB-relative addressing. |
| RND (Returned Data) | 10100 | The MCU option code RND enables the store field bus options (as in CRL) to store the U-bus in ACOR and initiate a "low-request" command. The DB-bank register generates the module number used to initiate a data fetch from memory. The returned data is loaded into the NIR register. |
| RNP | 10111 | Same as RND, except that the PB-bank register generates the module number. |
| RNS | 11100 | Same as RND, except that the stack-bank register generates the module number. |
| ROA | 00111 | Same as RND, except that the ABS-bank register generates the module number and the returned data is loaded into the OPND register. |

Table 3-9. MCU Option Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| ROD | 10111 | Same as RND, except that the DB-bank register generates the module number and the returned data is loaded into the OPND register. |
| ROND | 10011 | Same as RND, except that the returned data is stored in both the NIR and OPND registers. |
| RONP | 01011 | Same as RNP, except that the returned data is stored in both the NIR and OPND registers. |
| RONS | 11011 | Same as RNS, except that the returned data is stored in both the NIR and OPND registers. |
| ROP | 01111 | Same as RNP, except that the returned data is stored in the OPND register. |
| ROS | 11111 | Same as RNS, except that the returned data is stored in the OPND register. |
| ROSA | 00101 | Same as ROA, except that the addressed word is set to all 1's in memory during the same, non-interruptable, memory cycle. (Series II only.) |
| ROSD | 10101 | Same as ROSA, except that DB-bank is used. (Series II only.) |
| S | 11000 | Same as ABS, except that the stack-bank register is specified and is used with DB, Q, or S-relative addressing. |
| WRA | 00110 | The MCU option code WRA enables the store field bus options (as in CRL) to store the U-bus in ACOR and initiate a "low-request" command. The ABS-bank register generates the module number used to initiate a data store into memory. On the "select" cycle, the addressed memory module interprets the MCU bus data as an address and goes "busy". The module stays busy until it receives the data to be stored (normally sent on the following cycle with a microcode BUS DATA instruction) and completes the "write" cycle, or until its timer runs down. |
| WRD | 10110 | Same as WRA, except that the DB-bank register generates the module number. |
| WRS | 11110 | Same as WRA, except that the stack-bank register generates the module number. |

Table 3-6. Skip Field Code Definitions

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| BIT6 | 00101 | The skip field code BIT6 sets the NOP2 FF if bit 6 of the U-bus word is a logic 1. |
| BIT8 | 00110 | The skip field code BIT8 sets the NOP2 FF if bit 8 of the U-bus word is a logic 1. |
| CRRY (Carry) | 01000 | The skip field code CRRY sets the NOP2 FF if the ALU carry out is a logic 1. |
| CTRM (Counter Max) | 11011 | The skip field code CTRM sets the NOP2 FF if the counter contains all ones. |
| EVEN | 00010 | The skip field code EVEN sets the NOP2 FF if the U-bus word is an even number (U-bus bit 15 is a logic 0). |
| F1 (Flag 1) | 01100 | The skip field code F1 sets the NOP2 FF if Flag 1 FF is set. |
| F2 (Flag 2) | 01110 | The skip field code F2 sets the NOP2 FF if Flag 2 FF is set. |
| F3 (Flag 3) | 11100 | The skip field code F3 sets the NOP2 FF if Flag 3 FF is set. |
| INDR (Indirect) | 10100 | The skip field code INDR sets the NOP2 FF if the Indirect Bit FF is set and the indirect signal is a logic 1. |
| JLUI | 11001 | The skip field code JLUI causes a microjump to the ROM address specified by the LUT (look up table) providing the indirect condition (skip field INDR code) is not met. If the indirect condition is met the microjump is not executed. |
| NCRY | 01001 | The skip field code NCRY sets the NOP2 FF if the carry out from the ALU is zero. |
| NEG | 01011 | The skip field code NEG sets the NOP2 FF if the U-bus word is a negative number (U-bus bit 0 is a logic 1). |

Table 3-6. Skip Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| NEXT | 11101 | Terminates current instruction and initiates the sequence necessary to begin execution of the next instruction. If stackop A has just been executed and stackop B is not a NOP, then the hardware executes stackop B. Otherwise the action shown in the timing figure below takes place (a, b, c, d, e, f are equal length CPU clock cycles):<br><br>a b c d e f<br>... Mem. Sel. cycle DATA→NIR NIR→LUT ... NEXT: BUSL, RWP; Issue LOREQ; LUT→ VBUS→ ROM→ RANK1; NIR→CIR — NOP2: P+1→P Select cycle. RANK1→ RANK2 (if memory reference, force PADD, BASE to R, S-BUS Reg's.) — execute 1st line of micro code of new instr.<br><br>Time periods a, b, c (if present), and d occur in the currently executing instruction. "a" and "b" must occur before "d" for maximum execution speed – otherwise a CPU freeze will occur at "d". "a" and "b" result from the "next instruction prefetch" of the current instruction. "c" may or may not be present depending on the length of the current instruction. "d" is the last line of the current instruction. It initiates a "next instruction prefetch", transfers (NIR) to CIR, and applied the address on the VBUS (normally using the LUT output) to the ROM input. The ROM word at this address is stored in RANK1. In addition, the NOP2 FF is set. "e" is used to increment the P-reg., transfer RANK1 to RANK2, and if the new instruction is a memory-reference type, load the R- and S-BUS reg's. with the Pre-adder output and the proper base register. This is also the "select" cycle for the "next instr. prefetch" if there is no MCU conflict. During "f", the first line of the new instruction is executed.<br><br>The above is the normal sequence of operation of NEXT. This sequence is modified in the event an interrupt is pending or the micro code line is "... DATA NEXT".<br><br>"NEXT" also clears F1, F2, F3, CNTR, Subroutine Flag FF, and the ABS-BANK reg. |
| NF1 (Not Flag 1) | 01101 | The skip field code NF1 sets the NOP2 FF if Flag 1 FF is cleared. |
| NF2 (Not Flag 2) | 01111 | The skip field code NF2 sets the NOP2 FF if Flag 2 FF is cleared. |
| NOFL (Not Overflow) | 00111 | The skip field code NOFL sets the NOP2 FF if the ALU overflow bit is not a logic 1. Causes conditional jump and JSB to be two-cycle instructions. |
| NOP (No Operation) | 11111 | |

Table 3-6. Skip Field Code Definitions (Continued)

| LABEL AND NAME | FIELD CODE | DESCRIPTION |
|---|---|---|
| NPRV (Not Privileged) | 10110 | The skip field code NPRV sets the NOP2 FF if the privileged mode bit (status word bit 0) is zero. |
| NSME (Not Same) | 00100 | The skip field code NSME sets the NOP2 FF if all bits of the T-bus are not the same. |
| NZRO (Not Zero) | 00001 | The skip field code NZRO sets the NOP2 FF if the T-bus word is not equal to zero. |
| ODD | 00011 | The skip field code ODD sets the NOP2 FF if the U-bus word is an odd number (U-bus bit 15 is a logic 1). |
| POS (Positive) | 01010 | The skip field code POS sets the NOP2 FF if the U-bus word is a positive number (U-bus bit 0 is a logic 0). |
| RSB (Return from Subroutine) | 11000 | The skip field code RSB causes a microjump to the ROM address contained in the save register. |
| SR4 (SR=4) | 10010 | The skip field code SR4 sets the NOP2 FF if the SR register content is equal to four. |
| SRL2 (SR<2) | 10101 | The skip field code SRL2 sets the NOP2 FF if the SR register content is less than two. |
| SRL3 (SR<3) | 10111 | The skip field code SRL3 sets the NOP2 FF if the SR register content is less than three. |
| SRN4 (SR Not 4) | 10011 | The skip field code SRN4 sets the NOP2 FF if the SR register content is not four. |
| SRNZ (SR Not Zero) | 10001 | The skip field code SRNZ sets the NOP2 FF if the SR register content is not zero. |
| SRZ (SR Zero) | 10000 | The skip field code SRZ sets the NOP2 FF if the SR register content is equal to zero. |
| TEST | 11010 | The skip field code TEST sets the NOP2 FF if any enabled interrupt is pending. |
| UNC (Unconditional) | 11110 | The skip field code UNC and/or unconditional JMP's set the NOP2 FF. |
| ZERO | 00000 | The skip field code ZERO sets the NOP2 FF if the T-bus word is equal to zero. |