2. Index Register Organization

The index register can be addressed in two modes

a. By specifying 1 out of 16 possible locations with an OPA code of the form RRRR(1) (See Table III).

b. By specifying 1 out of 8 pairs with an OPA code of the form RRRX(2) (See Table III).

When the index register is used as a pair register, the even number register (RRR0) is used as the location of the middle address or the upper data fetched from the ROM, the odd number register (RRR1) is used as the location of the lower address or the lower data fetched from the ROM.

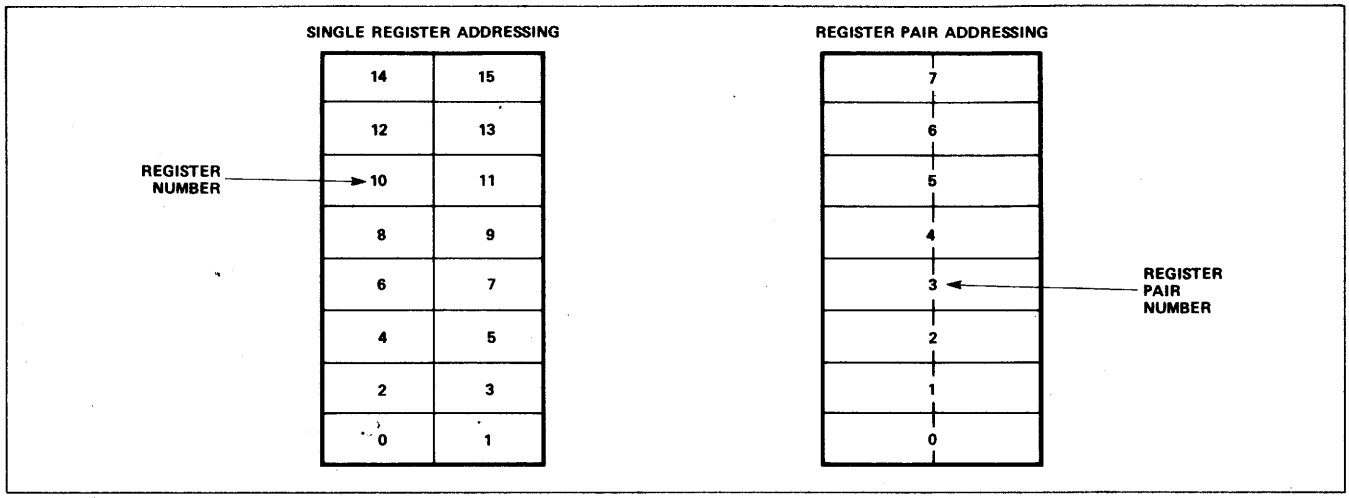


Table III - Index Register Organization

3. Operation of the Address Register (Program Counter and Stack)

The address register contains four 12-bit registers; one register is used as the program counter and stores the instruction address. the other 3 registers make up the push down stack.

Initially any one of the 4 registers can be used as the program counter to store the instruction address. In a typical sequence the program counter is incremented by 1 after the last address is sent out. This new address then becomes the effective address. If a JMS (Jump to Subroutine) instruction is received by the CPU, the program control is transferred to the address called out in JMS instruction. This address is stored in the register just above the old program counter which now saves the address of the next instruction to be executed following the last JMS.(3) This return address becomes the effective address following the BBL(Branch back and load) instruction at the end of the subroutine.

(1) In this case the instruction is executed on the 4-bit content addressed by RRRR.

(2) In this case the instruction is executed on the 8-bit content addressed by RRRX, where X is specified for each instruction.

(3) Since the JMS instruction is a 2-word instruction the old effective address is incremented by 2 to correctly give the address of the next instruction to be executed after the return from JMS.

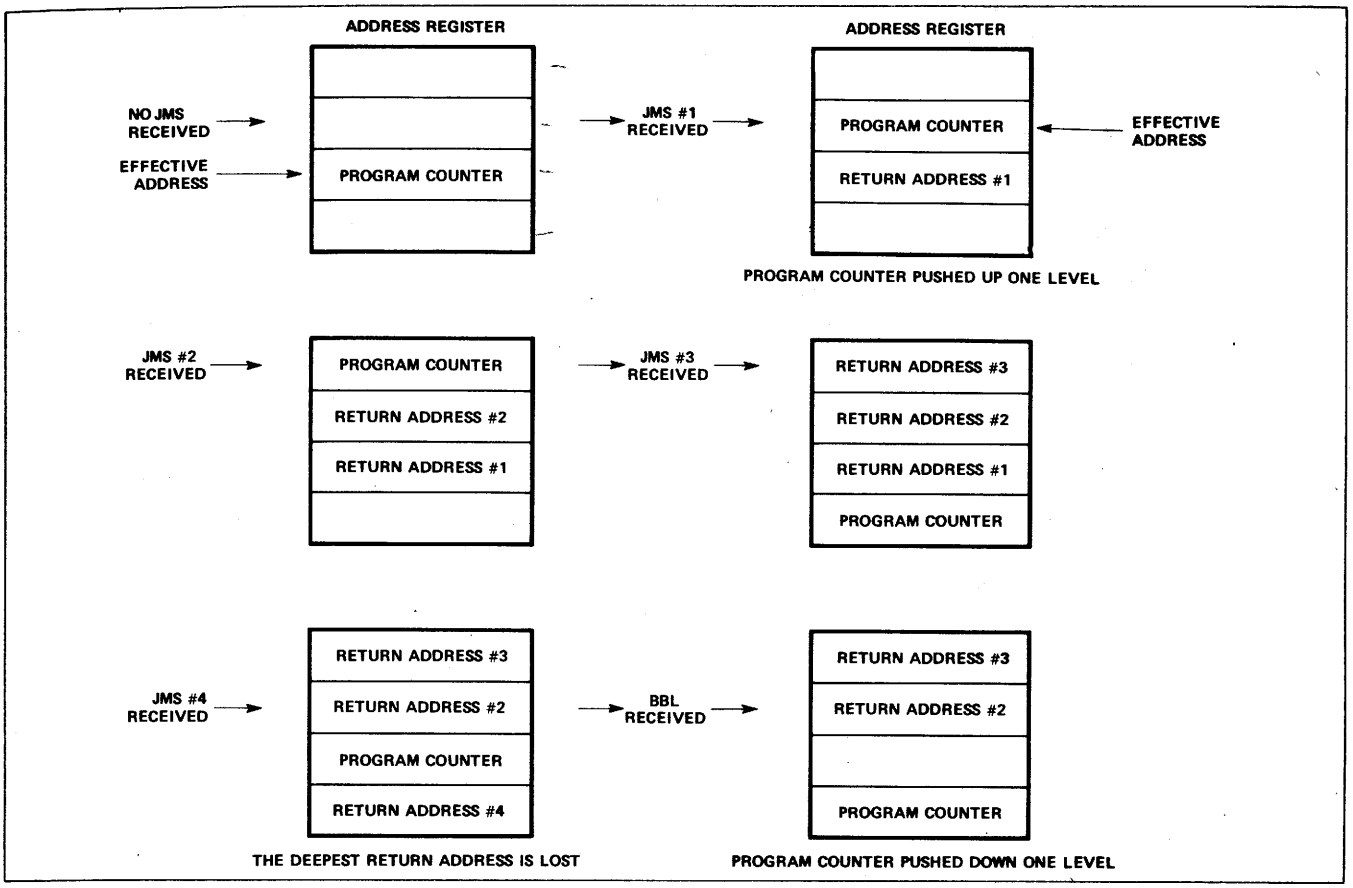


Table IV - Operation of the Address Register on a Jump to Subroutine Instruction

In summary, then, a JMS instruction pushes the program counter up one level and a BBL instruction pushes the program counter down one level. Since there are 3 registers in the push down stack, 3 return addresses may be saved. If a fourth JMS occurs, the deepest return address (the first one stored) is lost.

Table IV shows the operation of the address stack.

4. Operation of The Command Lines and the SRC Command

The CPU command lines (CM-ROM, CM-RAM_i) are used to control the ROM's and RAM's by indicating to them how to interpret the data bus content at any given time.

The command lines allow the implementation of RAM bank, chip, register and character addressing, ROM chip addressing, as well as activating the instruction control in each ROM and RAM chip at the time the CPU receives an I/O and RAM group instruction.

In a typical system configuration the CM-ROM line can control up to sixteen 4001's and each CM-RAM_i line can control up to four 4002's.

Each CM-RAM_i line can be selected by the execution of the DCL (Designate Command Line) instruction. The CM-ROM line, however, is always enabled.(1)

(1) If the number of ROM's in the system needs to be more than 16, external circuitry can be used to route CM-ROM to two ROM banks. The same comment applies to the CM-RAM_i lines if more than 16 RAM's need to be used.

Mnemonic: **FIN** (Fetch indirect from ROM)
OPR OPA: 0011 RRR0
Symbolic: (P_H) (0000) (0001) $\longrightarrow$ ROM address
(OPR) $\longrightarrow$ RRR0
(OPA) $\longrightarrow$ RRR1
Description: The 8 bit content of the 0 index register pair (0000) (0001) is sent out as an address in the same page where the FIN instruction is located. The 8 bit word at that location is loaded into the designated index register pair. The program counter is unaffected; after FIN has been executed the next instruction in sequence will be addressed. The content of the 0 index register pair is unaltered unless index register 0 was designated.

EXCEPTIONS:
a) Although FIN is a 1-word instruction, its execution requires two memory cycles (21.6 μsec).

b) When FIN is located at address (P_H) 1111 1111 data will be fetched from the next page(ROM) in sequence and not from the same page(ROM) where the FIN instruction is located. That is, next address is $(P_H + 1)$ (0000) (0001) and not (P_H) (0000) (0001).

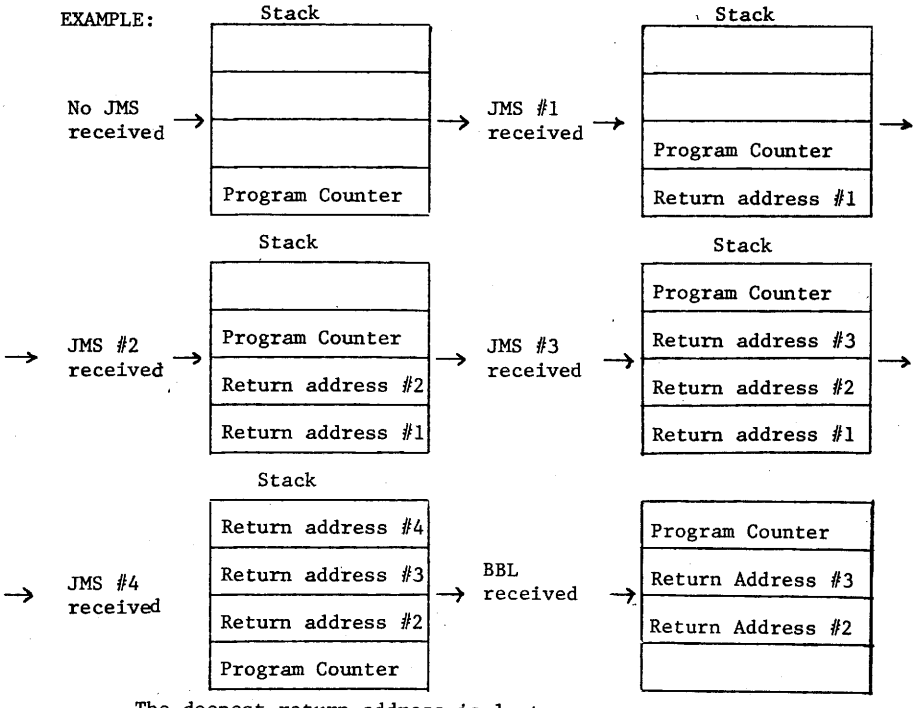
E. Two Word Machine Instruction

Mnemonic: **JUN** (Jump unconditional)
1st word OPR OPA: 0100 $A_3\ A_3\ A_3\ A_3$
2nd word OPR OPA: $A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1\ A_1$
Symbolic: $A_1\ A_1\ A_1\ A_1 \longrightarrow P_L$, $A_2\ A_2\ A_2\ A_2 \longrightarrow P_M$, $A_3\ A_3\ A_3\ A_3 \longrightarrow P_H$
Description: Program control is unconditionally transferred to the instruction locater at the address $A_3\ A_3\ A_3\ A_3$, $A_2\ A_2\ A_2\ A_2$, $A_1\ A_1\ A_1\ A_1$.

Mnemonic: **JMS** (Jump to Subroutine)
1st word OPR OPA: 0101 $A_3\ A_3\ A_3\ A_3$
2nd word OPR OPA: $A_2\ A_2\ A_2\ A_2$ $A_1\ A_1\ A_1$
Symbolic: $(P_H, P_M, P_L + 2) \longrightarrow$ Stack
$A_1\ A_1\ A_1\ A_1 \longrightarrow P_L$, $A_2\ A_2\ A_2\ A_2 \longrightarrow P_M$,
$A_3\ A_3\ A_3\ A_3 \longrightarrow P_H$
Description: The address of the next instruction in sequence following JMS (return address) is saved in the push down stack. Program control is transferred to the instruction located at the 12 bit address ($A_3A_3A_3A_3A_2A_2A_2A_2A_1A_1A_1A_1$). Execution of a return instruction (BBL) will cause the saved address to be pulled out of the stack, therefore, program control is transferred to the next sequential instruction after the last JMS.

The push down stack has 4 registers. One of them is used as the program counter, therefore nesting of JMS can occur up to 3 levels.

EXAMPLE:



The deepest return address is lost