Detailed Instruction Repertoire of the MCS-4

home

Intel 4004 Instructions Set

INSTRUCTION	MNEMONIC	BINARY EQUIVALENT 1st byte	2nd byte	MODIFIERS
No Operation	NOP	00000000	-	none
Jump Conditional	JCN	0001CCCC	AAAAAAAA	condition, address
Fetch Immediate	FIM	0010RRR0	DDDDDDDD	register pair, data
Send Register Control	SRC	0010RRR1	-	register pair
Fetch Indirect	FIN	0011RRR0	-	register pair
Jump Indirect	JIN	0011RRR1	-	register pair
Jump Uncoditional	JUN	0100AAAA	AAAAAAAA	address
Jump to Subroutine	JMS	0101AAAA	AAAAAAAA	address
Increment	INC	0110RRRR	-	register
Increment and Skip	ISZ	0111RRRR	AAAAAAAA	register, address
Add	ADD	1000RRRR	-	register
Subtract	SUB	1001RRRR	-	register
Load	LD	1010RRRR	-	register
Exchange	XCH	1011RRRR	-	register
Branch Back and Load	BBL	1100DDDD	-	data
Load Immediate	LDM	1101DDDD	-	data
Write Main Memory	WRM	11100000	-	none
Write RAM Port	WMP	11100001	-	none
Write ROM Port	WRR	11100010	-	none
Write Status Char 0	WR0	11100100	-	none
Write Status Char 1	WR1	11100101	-	none
Write Status Char 2	WR2	11100110	-	none
Write Status Char 3	WR3	11100111	-	none
Subtract Main Memory	SBM	11101000	-	none
Read Main Memory	RDM	11101001	-	none
Read ROM Port	RDR	11101010	-	none
Add Main Memory	ADM	11101011	-	none
Read Status Char 0	RD0	11101100	-	none
Read Status Char 1	RD1	11101101	-	none
Read Status Char 2	RD2	11101110	-	none
Read Status Char 3	RD3	11101111	-	none
Clear Both	CLB	11110000	-	none
Clear Carry	CLC	11110001	-	none
Increment Accumulator	IAC	11110010	-	none
Complement Carry	CMC	11110011	-	none
Complement	CMA	11110100	-	none
Rotate Left	RAL	11110101	-	none
Rotate Right	RAR	11110110	-	none
Transfer Carry and Clear	TCC	11110111	-	none
Decrement Accumulator	DAC	11111000	-	none
Transfer Carry Subtract	TCS	11111001	-	none
Set Carry	STC	11111010	-	none
Decimal Adjust Accumulator	DAA	11111011	-	none
Keybord Process	KBP	11111100	-	none
Designate Command Line	DCL	11111101	-	none

Instuction Format

The MCS-4 micro computer set has two types of instuction
a) 1 word instruction with an 8-bit code and an execution time of 10.8 usec.
b) 2 word instruction with an 16-bit code and an execution time of 21.6 usec.

Due to the time multiplexed operation of the system, the 8-bit instruction is fetched 4-bits at a time on two succesive clock periods.

The instruction formats are illustrated in Tables I and II.

Table I

ONE WORD INSTRUCTION

D3	D2	D1	D0	D3	D2	D1	D0
x	x	x	x	x	x	x	x
OPR				OPA			

OP CODE	MODIFIER

x	x	x	x	INDEX REGISTER ADDRESS R R R R

OR

x	x	x	x	INDEX REGISTER PAIR ADDRESS R R R X

OR

x	x	x	x	DATA D D D D

Table II

TWO WORD INSTRUCTIONS

1st INSTRUCTION CYCLE

D3	D2	D1	D0	D3	D2	D1	D0
x	x	x	x	x	x	x	x
OPR				OPA			

OP CODE	MODIFIER

x	x	x	x	UPPER ADDRESS A3 A3 A3 A3

OR

x	x	x	x	CONDITION C1 C2 C3 C4

OR

x	x	x	x	INDEX REGISTER ADDRESS R R R R

OR

x	x	x	x	INDEX REGISTER PAIR ADDRESS R R R X

2nd INSTRUCTION CYCLE

D3	D2	D1	D0	D3	D2	D1	D0
x	x	x	x	x	x	x	x
OPR				OPA			

OP CODE	MODIFIER

MIDDLE ADDRESS A2 A2 A2 A2				LOWER ADDRESS A1 A1 A1 A1				

OR

MIDDLE ADDRESS A2 A2 A2 A2				LOWER ADDRESS A1 A1 A1 A1				

OR

MIDDLE ADDRESS A2 A2 A2 A2				LOWER ADDRESS A1 A1 A1 A1				

OR

UPPER DATA D2 D2 D2 D2				LOWER DATA D1 D1 D1 D1				

Symbols and Abbreviations

The following Symbols and abbreviations will be used thorughout the next few sections:

( )	the content of
-->	is transferred to
ACC	Accumulator (4-bit)
CY	Carry/link Flip-Flop
ACBR	Accumulator Buffer Register (4-bit)
RRRR	Index register address
RRR	Index register pair address
PL	Low order program counter Field (4-bit)
PM	Middle order program counter Field (4-bit)
PH	High order program counter Field (4-bit)
ai	Order i content of the accumulator
CMi	Order i content of the command register
M	RAM main character location

MSi RAM status character i
DB (T) Data bus content at time T
Stack The 3 registers in the address register other than the program counter

Format for Describing Each Instruction

Each instruction will be described as follows:
(1) Mnemonic symbol and meaning
(2) OPR and OPA code
(3) Symbolic representation of the instruction
(4) Description of the instruction (if necessary)
(5) Example and/or exceptions (if necessary)

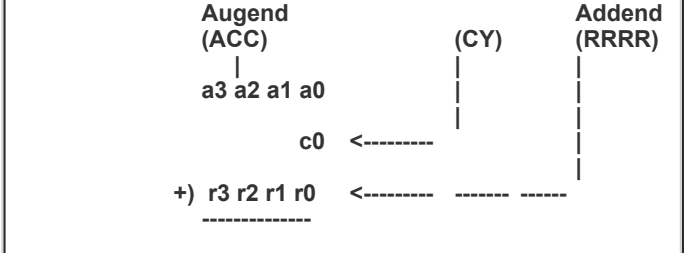
One Word Machine Instruction

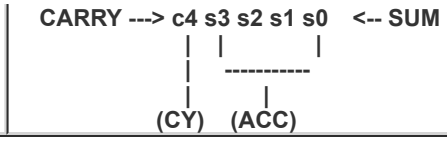
Mnemonic:	**NOP (No Operation)**
OPR OPA:	**0000 0000**
Symbolic:	**Not applicable**
Description:	**No operation performed**

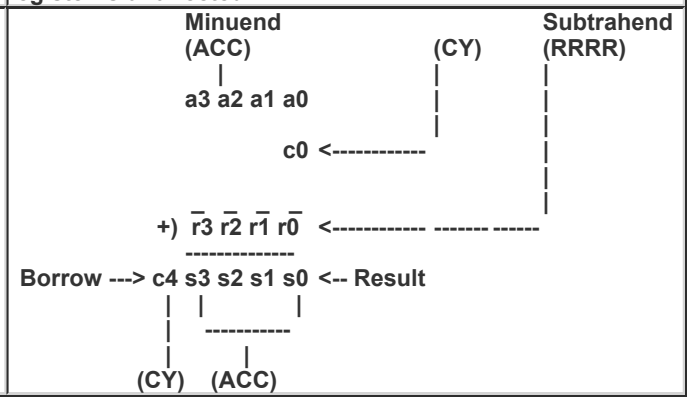
Mnemonic:	**LDM (Load data to Accumulator)**
OPR OPA:	**1101 DDDD**
Symbolic:	**DDDD --> ACC**
Description:	**The 4 bits of data, DDDD stored in the OPA field of insruction word are loaded into the accumulator. The previous contents of the acummulator are lost. The carry/link bit is unaffected.**

Mnemonic:	**LD (Load index register to Accumulator)**
OPR OPA:	**1010 RRRR**
Symbolic:	**(RRRR) --> ACC**
Description:	**The 4 bit content of the designated index register (RRRR) is loaded into accumulator. The previous contents of the accumulator are lost. The 4 bit content of the index register and the carry/link bit are unaffected.**

Mnemonic:	**XCH (Exchange index register and accumulator)**
OPR OPA:	**1011 DDDD**
Symbolic:	**(ACC) --> ACBR, (RRRR) --> ACC, (ACBR) --> RRRR**
Description:	**The 4 bit content of designated index register is loaded into the accumulator. The prior content of the accumulator is loaded into the designed register. The carry/link bit is unaffected.**

Mnemonic:	**ADD (Add index register to accumulator with carry)**
OPR OPA:	**1000 RRRR**
Symbolic:	**(RRRR) + (ACC) + (CY) --> ACC, CY**
Description:	**The 4 bit content of the designated index register is added to the content of the accumulator with carry. The result is stored in the accumulator. The carry/link is set to 1 if a sum greater than 15 was generated to indicate a carry out; otherwise, the carry/link is set to 0. The 4 bit content of the index register is unaffected.**
Example:	

Mnemonic:	**SUB (Subtract index register from accumulator with borrow)**
OPR OPA:	**1001 RRRR**
Symbolic:	**(ACC) + ~(RRRR) + (CY) --> ACC, CY**
Description:	**The 4 bit content of the designated index register is complemented (ones complement) and added to content of the accumulator with borrow and the result is stored in the accumulator. If a borrow is generated, the carry bit is set to 0; otherwise, it is set to 1. The 4 bit content of the index register is unaffected.**
Example:	

Mnemonic:	**INC (Increment index register)**
OPR OPA:	**0110 RRRR**
Symbolic:	**(RRRR) +1 --> RRRR**
Description:	**The 4 bit content of the designated index register is incremented by 1. The index register is set to zero in case of overflow. The carry/link is unaffected.**

Mnemonic:	**BBL (Branch back and load data to the accumulator)**
OPR OPA:	**0110 RRRR**
Symbolic:	**(Stack) --> PL, PM, PH; DDDD --> ACC**
Description:	**The program counter (address stack) is pushed down one level. Program control transfers to the next instruction following the last jump to subroutine (JMS) instruction. The 4 bits of data DDDD stored in the OPA portion of the instruction are loaded to the accumulator.** **BBL is used to return from subroutine to main program.**

Mnemonic:	**JIN (Jump indirect)**
OPR OPA:	**0011 RRR1**
Symbolic:	**(RRRO) --> PM** **(RRR1) --> PL; PH unchanged**
Description:	**The 8 bit content of the designated index register pair is loaded into the low order 8 positions of the program counter. Program control is transferred to the instruction at that address on the same page (same ROM) where the JIN instruction is located. The 8 bit content of the index register is unaffected.**
EXCEPTIONS:	**When JIN is located at the address (PH) 1111 1111 program control is transferred to the next page in sequence and not to the same page where the JIN instruction is located. That is, the next address is (PH + 1) (RRRO) (RRR1) and not (PH) (RRRO) (RRR1)**

Mnemonic:	**SRC (Send register control)**
OPR OPA:	**0010 RRR1**

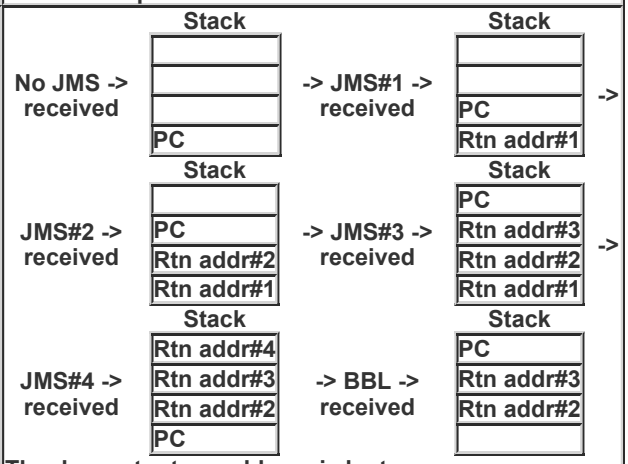
Symbolic:	**(RRRO) --> DB (X2) (RRR1) --> DB (X3)**
Description:	**The 8 bit content of the designated index register pair is sent to the RAM address register at X2 and X3. A subsequent read, write, or I/O operation of the RAM will utilize this address. Specifically, the first 2 bits of the address designatea RAM chip; the second 2 bits designate 1 out of 4 registers within the chip; the last 4 bits designate 1 out of 16 4-bit main memory characters within the register. This command is also used to designate a ROM for a subsequent ROM I/O port operation. The first 4 bits designate the ROM chip number to be selected. The address in ROM or RAM is not cleared until the next SRC instruction is executed. The 8 bit content of the index register is unaffected.**

Mnemonic:	**FIN (Fetch indirect from ROM)**
OPR OPA:	**0011 RRRO**
Symbolic:	**(PH) (0000) (0001) --> ROM address (OPR) --> RRRO (OPA) --> 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 psec). b) When FIN is located at address (PH) 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 (PH + 1) (0000) (0001) and not (PH) (0000) (0001).**

Two Word Machine Instruction

Mnemonic:	**JUN (Jump unconditional)**
1st word OPR OPA:	**0100 A3 A3 A3 A3**
2nd word OPR OPA:	**A2 A2 A2 A2 AI AI AI AI**
Symbolic:	**AI AI AI AI --> PL, A2 A2 A2 A2 --> PM, A3 A3 A3 A3 --> PH**
Description:	**Program control is unconditionally transferred to the instruction locater at the address A3 A3 A3 A3, A2 A2 A2 A2, AI AI AI AI.**

Mnemonic:	**JMS (Jump to Subroutine)**
1st word OPR OPA:	**0101 A3 A3 A3 A3**
2nd word OPR OPA:	**A2 A2 A2 A2 A1 A1 A1 A1**
Symbolic:	**AI AI AI AI --> PL, A2 A2 A2 A2 --> PM, A3 A3 A3 A3 --> PH**
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 (A3A3A3A3A2A2A2A2A1A1A1A1). 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.

Mnemonic:	JCN (Jump conditional)
1st word OPR OPA:	0001 C1C2C3C4
2nd word OPR OPA:	A2 A2 A2 A2 A1 A1 A1 A1
Symbolic:	If C1C2C3C4 is true, A2A2A2A2 --> PM A1A1A1A1 --> PL, PH unchanged if C1C2C3C4 is false, (PH) --> PH, (PM) --> PM, (PL + 2) --> PL
Description:	If the designated condition code is true, program control is transferred to the instruction located at the 8 bit address A2A2A2A2, A1A1A1A1 on the same page (ROM) where JCN is located. If the condition is not true the next instruction in sequence after JCN is executed. The condition bits are assigned as follows: C1 = 0 Do not invert jump condition C1 = 1 Invert jump condition C2 = 1 Jump if the accumulator content is zero C3 = 1 Jump if the carry/link content is 1 C4 = 1 Jump if test signal (pin 10 on 4004) is zero.
Example:	OPR OPA ------ ------ 0001 0110 Jump if accumulator is zero or carry = 1 Several conditions can be tested simutaneously. The logic equation describing the condition for a jump is give below: JUMP = ~C1 . ((ACC = 0) . C2 + (CY = 1) . C3 + ~TEST . C4) + C1 . ~((ACC = 0) . C2 + (CY = 1) . C3 + ~TEST . C4)
EXCEPTIONS:	If JCN is located on words 254 and 255 of a ROM page, when JCN is executed and the condition is true, program control is transferred to the 8-bit address on the next page where JCN is located.

Mnemonic:	ISZ (Increment index register skip if zero)
1st word OPR OPA:	0111 RRRR
2nd word OPR OPA:	A2 A2 A2 A2 A1 A1 A1 A1
Symbolic:	(RRRR) + 1 --> RRRR, if result = 0 (PH) --> PH, (PM) --> PM, (PL + 2) --> PL: if result <> 0 (PH) --> PH, A2A2A2A2 --> PM, A1A1A1A1 --> PL
Description:	The content of the designated index register is incremented by 1. The accumulator and carry/link are

	unaffected. If the result is zero, the next instruction after ISZ is executed. If the result is different from 0, program control is transferred to the instruction located at the 8 bit address A2A2A2A2, A1A1A1A1 on the same page (ROM) where the ISZ instruction is located.
EXCEPTIONS:	If ISZ is located on words 254 and 255 of a ROM page, when ISZ is executed and the result is not zero, program control is transferred to the 8-bit address located on the next page in sequence and not on the same page where ISZ is located.

Mnemonic:	FIM (Fetched immediate from ROM)
1st word OPR OPA:	0010 RRR0
2nd word OPR OPA:	D2 D2 D2 D2 D1 D1 D1 D1
Symbolic:	D2D2D2D2 --> RRR0, D1D1D1D1 --> RRR1
Description:	The 2nd word represents 8-bits of data which are loaded into the designated index register pair.

Input/Output and RAM Instructions

(The RAM's and ROM's operated on in the I/O and RAM instructions have been previously selected by the last SRC instruction executed.)

Mnemonic:	RDM (Read RAM character)
OPR OPA:	1110 1001
Symbolic:	(M) --> ACC
Description:	The content of the previously selected RAM main memory character is transferred to the accumulator. The carry/linkis unaffected. The 4-bit data in memory is unaffected.

Mnemonic:	RD0 (Read RAM status character 0)
OPR OPA:	1110 1100
Symbolic:	(MS0) --> ACC
Description:	The 4-bits of status character 0 for the previously selected RAM register are transferred to the accumulator. The carry/link and the status character are unaffected.

Mnemonic:	RD1 (Read RAM status character 1)
OPR OPA:	1110 1101
Symbolic:	(MS1) --> ACC

Mnemonic:	RD2 (Read RAM status character 2)
OPR OPA:	1110 1110
Symbolic:	(MS2) --> ACC

Mnemonic:	RD3 (Read RAM status character 3)
OPR OPA:	1110 1111
Symbolic:	(MS0) --> ACC

Mnemonic:	RDR (Read ROM port)
OPR OPA:	1110 1010
Symbolic:	(ROM input lines) --> ACC
Description:	The data present at the input lines of the previously selected ROM chip is transferred to the accumulator. The carry/link is unaffected. If the I/O option has both inputs and outputs within the same 4 I/O lines, the user can choose to have either "0" or "1" transferred to the accumulator for those I/O pins coded as outputs, when an RDR instruction is executed.
EXAMPLE:	Given a 4001 with I/O coded with 2 inputs and 2 outputs,

	when RDR is executed the transfer is as shown below: I3 O2 O1 I0 (ACC) 1 X X 0 1 --> 1 (1 or 0) (1 or 0) 0 Input Data User can choose

Mnemonic:	WRM (Write accumulator into RAM character)
OPR OPA:	1110 0000
Symbolic:	(ACC) --> M
Description:	The accumulator content is written into the previously selected RAM main memory character location. The accumulator and carry/link are unaffected.

Mnemonic:	WRO (Write accumulator into RAM status character 0)
OPR OPA:	1110 0100
Symbolic:	(ACC) --> MS0
Description:	The content of the accumulator is written into the RAM status character 0 of the previously selected RAM register. The accumulator and the carry/link are unaffected.

Mnemonic:	WR1 (Write accumulator into RAM status character 1)
OPR OPA:	1110 0101
Symbolic:	(ACC) --> MS1

Mnemonic:	WR2 (Write accumulator into RAM status character 2)
OPR OPA:	1110 0110
Symbolic:	(ACC) --> MS2

Mnemonic:	WR3 (Write accumulator into RAM status character 3)
OPR OPA:	1110 0111
Symbolic:	(ACC) --> MS3

Mnemonic:	WRR (Write ROM port)
OPR OPA:	1110 0010
Symbolic:	(ACC) --> ROM output lines
Description:	The content of the accumulator is transferred to the ROM output port of the previously selected ROM chip. The data is available on the output pins until a new WRR is executed on the same chip. The ACC content and carry/link are unaffected. (The LSB bit of the accumulator appears on I/O0, pin 16, of the 4001). No operation is performed on I/O lines coded as inputs.

Mnemonic:	WMP (Write memory port)
OPR OPA:	1110 0001
Symbolic:	(ACC) --> RAM output register
Description:	The content of the accumulator is transferred to the RAM output port of the previously selected RAM chip. The data is available on the output pins until a new WMP is executed on the same RAM chip. The content of the ACC and the carry/link are unaffected. (The LSB bit of the accumultor appears on O0, Pin 16, of the 4002.)

Mnemonic:	ADM (Add from memory with carry)
OPR OPA:	1110 1011
Symbolic:	(M) + (ACC) + (CY) --> ACC, CY
Description:	The content of the previously selected RAM main memory character is added to the accumulator with carry. The RAM character is unaffected.

Mnemonic:	SBM (Subtract from memory with borrow)
OPR OPA:	1110 1000

Symbolic:	~(M) + (ACC) + ~(CY) --> ACC, CY
Description:	Thecontent of the previously selected RAM character is subtracted from the accumulator with borrow. The RAM character is unaffected.

Accumulator Group Instructions

Mnemonic:	CLB (Clear both)
OPR OPA:	1111 0000
Symbolic:	0 --> ACC, 0 --> CY
Description:	Set accumulator and carry/link to 0.

Mnemonic:	CLC (Clear carry)
OPR OPA:	1111 0001
Symbolic:	0 --> CY
Description:	Set carry/link to 0.

Mnemonic:	CMC (Complement carry)
OPR OPA:	1111 0011
Symbolic:	~(CY) --> CY
Description:	The carry/link content is complemented.

Mnemonic:	STC (Set carry)
OPR OPA:	1111 1010
Symbolic:	1 --> CY
Description:	Set carry/link to a 1

Mnemonic:	CMA (Complement Accumulator)
OPR OPA:	1111 0100
Symbolic:	~a3 ~a2 ~a1 ~a0 --> ACC
Description:	The content of the accumulator is complemented. The carry/link is unaffected.

Mnemonic:	IAC (Increment accumulator)
OPR OPA:	1111 0010
Symbolic:	(ACC) +1 --> ACC
Description:	The content of the accumulator is incremented by 1. No overflow sets the carry/link to 0; overflow sets the carry/link to a 1.

Mnemonic:	DAC (decrement accumulator)
OPR OPA:	1111 1000
Symbolic:	(ACC) - 1 --> ACC
Description:	The content of the accumulator is decremented by 1. A borrow sets the carry/link to 0; no borrow sets the carry/link to a 1.
EXAMPLE:	(ACC) | a3 a2 a1 a0 +) 1 1 1 1 ---------- C4 S3 S2 S1 S0 | | CY ACC

Mnemonic:	RAL (Rotate left)
OPR OPA:	1111 0101
Symbolic:	C0 --> a0, a(i) --> a(i+1), a3 -->CY
Description:	The content of the accumulator and carry/link are rotated

	left.

Mnemonic:	RAR (Rotate right)
OPR OPA:	1111 0110
Symbolic:	a0 --> CY, a(i) --> a(i-1), C0 -->a3
Description:	The content of the accumulator and carry/link are rotated right.

Mnemonic:	TCC (Transmit carry and clear)
OPR OPA:	1111 0111
Symbolic:	0 --> ACC, (CY) --> a0, 0 --> CY
Description:	The accumulator is cleared. The least significant position of the accumulator is set to the value of the carry/link. The carry/link is set to 0.

Mnemonic:	DAA (Decimal adjust accumulator)
OPR OPA:	1111 1011
Symbolic:	(ACC) + (0000 or 0110) --> ACC
Description:	The accumulator is incremented by 6 if either the carry/link is 1 or if the accumulator content is greater than 9. The carry/link is set to a 1 if the result generates a carry, otherwise it is unaffected.

Mnemonic:	TCS (Transfer carry subtract)
OPR OPA:	1111 1001
Symbolic:	1001 --> ACC if (CY) = 0 1010 --> ACC if (CY) = 1 0 --> CY
Description:	The accumulator is set to 9 if the carry/link is 0. The accumulator is set to 10 if the carry/link is a 1. The carry/link is set to 0.

Mnemonic:	KBP (Keyboard process)
OPR OPA:	1111 1100
Symbolic:	(ACC) --> KBP, ROM --> ACC
Description:	A code conversion is performed on the accumulator content, from 1 out of n to binary code. If the accumulator content has more than one bit on, the accumulator will be set to 15 (to indicate error). The carry/link is unaffected. The conversion table is shown below:

(ACC) before KBP		(ACC) afer KBP
0000	---->	0000
0001	---->	0001
0010	---->	0010
0100	---->	0011
1000	---->	0100
0011	---->	1111
0101	---->	1111
0110	---->	1111
0111	---->	1111
1001	---->	1111
1010	---->	1111
1011	---->	1111
1100	---->	1111
1101	---->	1111
1110	---->	1111
1111	---->	1111

Mnemonic:	DCL (Designate command line)
OPR OPA:	1111 1101
Symbolic:	a0 --> CM0, a1 --> CM1, a2 --> CM2
Description:	The content of the three least significant accumulator bits is transferred to the comand control register within the CPU.

This instruction provides RAM bank selection when multiple RAM banks are used.(If no DCL instruction is sent out, RAM Bank number zero is automatically selected after application of at lease one RESET). DCL remains latched until it is changed.

The selection is made according to the following truth table.

(ACC)	CM-RAMi Enabled	Bank No.
X000	CM-RAM0	Bank 0
X001	CM-RAM1	Bank 1
X010	CM-RAM2	Bank 2
X100	CM-RAM3	Bank 3
X011	CM-RAM1, CM-RAM1	Bank 4
X101	CM-RAM1, CM-RAM3	Bank 5
X110	CM-RAM2, CM-RAM3	Bank 6
X111	CM-RAM1, CM-RAM2, CM-RAM3	Bank 7