INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONES: 224281/2/3/4/5/6
CABLE: CENTRATOM - TELEX 460392-1

SMR/90 - 7

COLLEGE ON MICROPROCESSORS:

TECHNOLOGY AND APPLICATIONS IN PHYSICS

7 September - 2 October 1981

MICROPROCESSOR INSTRUCTION SETS

M.D. EDWARDS

Computation Dept.
U.M.I.S.T.
Manchester
England

MICROPROCESSOR INSTRUCTION SETS

DR. MARTYN D. EDWARDS
COMPUTATION DEPT.,
U.M.I.S.T.,
MANCHESTER,
ENGLAND
U.K.

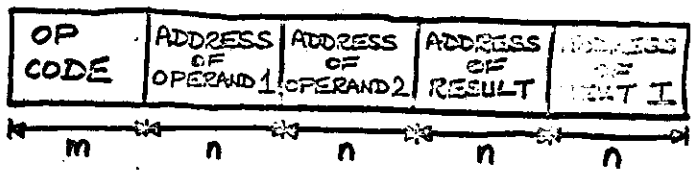
- OBJECTIVE
  TO LOOK AT THE REQUIREMENTS OF A μP INSTRUCTION SET.
  EXAMINE A TYPICAL 8-BIT μP
  M6800

- INSTRUCTION SETS
  - INSTRUCTION FORMAT
  - ADDRESSING METHODS
    - INSTRUCTIONS
    - OPERANDS
  - INSTRUCTION TYPES

INSTRUCTION FORMAT

- THE μP ACTIONS ARE DEFINED BY INSTRUCTIONS
- A SET OF INSTRUCTIONS IS A PROGRAM
- THE COLLECTION OF POSSIBLE INSTRUCTIONS IS KNOWN AS AN INSTRUCTION SET

AN INSTRUCTION DEFINES
- OPERATION (OPCODE)
- SOURCE OF THE DATA
- DESTINATION OF THE RESULT
- SOURCE OF THE NEXT INSTRUCTION



LENGTH OF INSTRUCTION (m+4n) bits

Eg : m = 4 (16 instructions)
n = 12 (12 bit address: 4096)
instruction = 52 bits long

- confined to 4, 8, 16 bit μP's
- very conservative m & n

■ HAVE TO REDUCE Instr. LENGTH
- HAVE A PC
- MAKE SOURCE/DESTINATION ADDRESSES IMPLICIT
- MAKE DESTINATION ADDRESS = ADDRESS OF ONE OF THE SOURCE OPERANDS
- LIMIT ADDRESSES TO BE ON-CHIP REGISTERS OR CONTENTS OF REGISTERS

- BY REDUCING THE LENGTH OF AN INSTRUCTION THEN FLEXIBILITY IS LOST.
- EXTRA INSTRUCTIONS ARE NECESSARY.

■ COMPUTERS WITH SHORT WORD LENGTHS GENERALLY USE AN ACCUMULATOR AS THE SOURCE AND DESTINATION OF OPERANDS
- KNOWN AS ONE-ADDRESS INSTRUCTIONS

- OPERAND SOURCE/DESTINATION IS EITHER
  - ACCUMULATOR(S)
  - MEMORY
  - ON-CHIP REGISTERS

- SOME INSTRUCTIONS HAVE AN IMPLIED SOURCE/DESTINATION OPERAND ADDRESSES → ACC

ADDRESSING METHODS

- WAY OF SPECIFYING A MEMORY ADDRESS THAT AN INSTRUCTION USES TO ACCESS OPERANDS
- ADDRESSING METHODS MUST BE
  - SIMPLE
  - SHORT ADDRESSES IN THE INSTRUCTION
  - FAST ACCESS TO THE OPERANDS
- WIDELY USED ADDRESSING TECHNIQUES
  - DIRECT
  - INDIRECT
  - IMMEDIATE
  - INDEXED
  - RELATIVE
  - REGISTER DIRECT
  - REGISTER INDIRECT

- DIRECT ADDRESSING
  $A \leftarrow A + M(\text{operand address})$
- INDIRECT ADDRESSING
  $A \leftarrow A + M((\text{operand address}))$
- IMMEDIATE ADDRESSING
  $A \leftarrow A + \text{operand}$
- INDEXED ADDRESSING
  $A \leftarrow A + M(\text{operand} + IX)$
- RELATIVE ADDRESSING
  $A \leftarrow A + M(\text{operand} + PC)$
- REGISTER DIRECT
  $A \leftarrow A + (\text{Register})$
- REGISTER INDIRECT
  $A \leftarrow A + M((\text{Register}))$

where A = accumulator
IX = index register
PC = program counter
M( ) = contents of
M(( )) = contents of the contents of

INSTRUCTION TYPES

- DATA MANIPULATION : TRANSFORMS DATA
  - ARITHMETIC
  - LOGICAL
  - SHIFT
  - COMPARISON
- DATA TRANSFER : MOVES DATA
  - MEMORY TRANSFER
  - I/O
  - INTERNAL TRANSFER
  - "STACK"
- PROGRAM CONTROL : CHANGES PC
  - UNCONDITIONAL JUMP
  - CONDITIONAL JUMP
  - SUBROUTINES

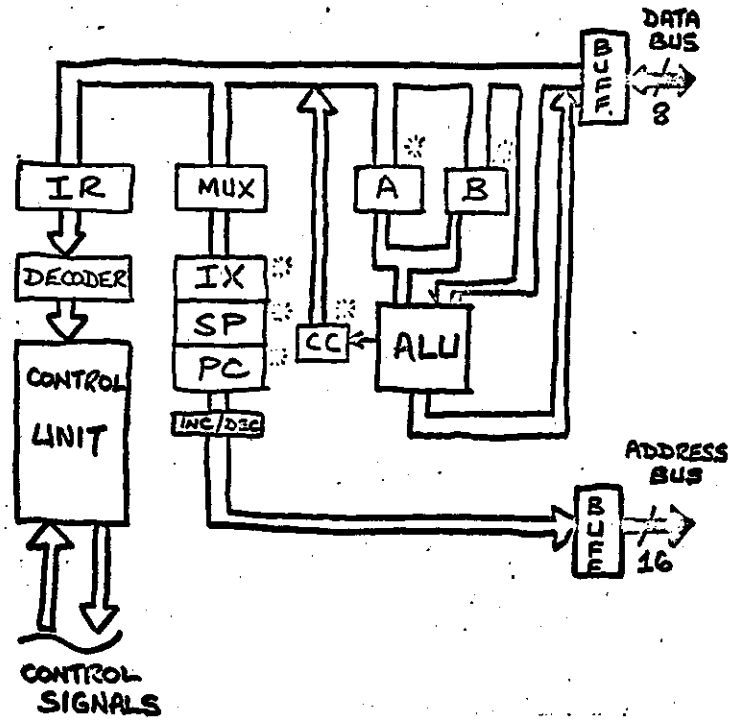
INSTRUCTION SETS (IS)

- No computer has every possible instruction
- Large IS → shorter programs, higher speed
- Large IS → longer instructions, more decoding
- µP's : 40-80 distinct instructions
- µP's : restricted addressing modes
- µP's : multibyte instructions

1 byte, 2 byte, 3 byte instructions are common.

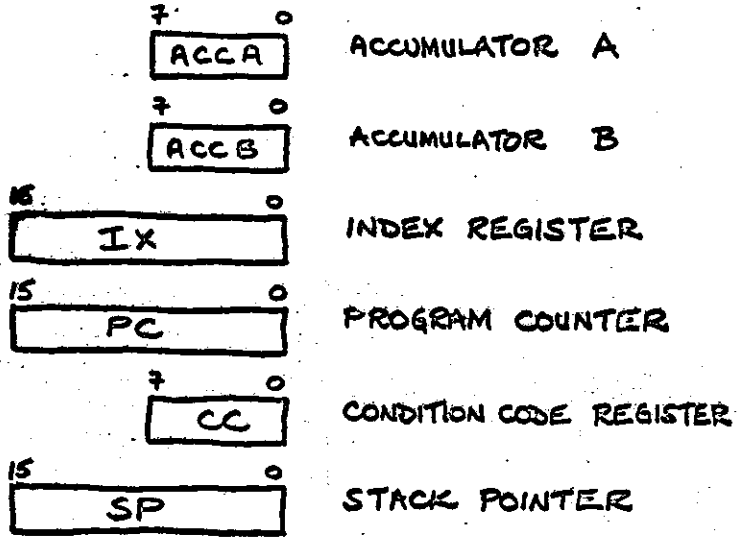
▌1-ADDRESS▯ nominally

M6800 ARCHITECTURE



- * INDICATES INTERNAL REGISTERS ACCESSIBLE BY THE PROGRAMMER

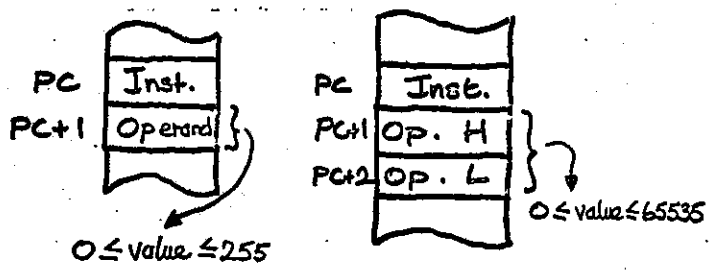
REGISTER STRUCTURE



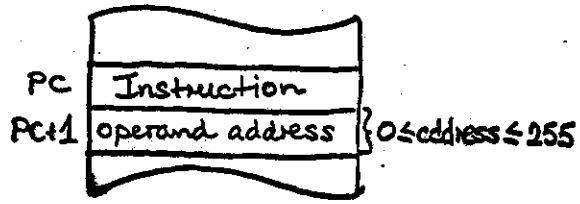
- IX register is used in the 'INDEXED ADDRESSING' MODE of the 6800.
- SP is used to access an 'operand stack' and for controlling SUBROUTINE access.

OPERAND ADDRESSING

- IMMEDIATE MODE
  - the operand field of the instruction contains the operand itself and is located in the byte (8-bit operand) or bytes (16-bit operand) following the instruction.

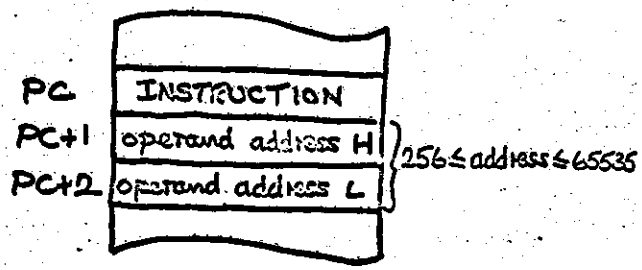


- DIRECT MODE
  - the operand field of the instruction contains the address of the operand. Memory locations in the range 0 → 255 may be accessed.



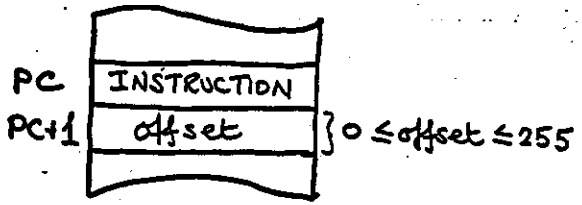
EXTENDED MODE

- identical to the direct mode except that a 16-bit operand address field is required.



INDEXED MODE

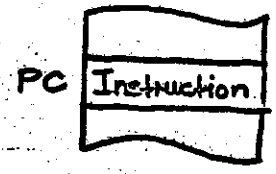
- Similar to direct addressing except that the operand address is 'variable'. The operand field is added to the contents of the INDEX REGISTER to find the address of the operand.



- operand address = (IX) + offset

IMPLIED MODE

- the address of the operand is inherent in the instruction itself and normally refers to one of the processor registers.



SUMMARY

IMMEDIATE : (R) ← (R) b (n) ; (R) ← (n)

DIRECT : (R) ← (R) b M(n) ; (R) ← M(n)
M(n) ← (R)

EXTENDED : (R) ← (R) b M(n) ; (R) ← M(n)
M(n) ← (R) ; M(n) ← M(n) u

INDEXED : (R) ← (R) b M(IX+n) ; (R) ← M(IX+n)
M(IX+n) ← (R) ; (R) ← M(IX+n)
M(IX+n) ← M(IX+n) u

IMPLIED : (R1) ← (R1) b (R2) ; (R) ← (R) u
(R1) ← (R2)

- R, R1, R2 : internal registers ; n : operand
M(n), M(IX+n) : memory location
b = binary operator ; u = unary operator

MOVE OPERATORS

- operations merely copy the contents of the source operand, X, to the destination operand, Z.
- $(R) \leftarrow (n)$; $(R) \leftarrow M(n)$; $(R) \leftarrow M(IX+n)$
  $M(IX+n) \leftarrow (R)$; $M(n) \leftarrow (R)$; $(R1) \leftarrow (R2)$

		In	D	E	Ix	I.P
• LDAA	: LOAD A ACC	•	•	•	•	
LDAB	: LOAD B ACC	•	•	•	•	
STAA	: STORE A ACC		•	•	•	
STAB	: STORE B ACC		•	•	•	
TAB	: (B) ← (A)					•
TBA	: (A) ← (B)					•
LDX	: LOAD IX	•	•	•	•	
LDS	: LOAD SP	•	•	•	•	
STX	: STORE IX		•	•	•	
STS	: STORE SP		•	•	•	

8-BIT OPERATORS

- OPERATIONS PERFORMED ON 8-BIT OPERANDS
- MAJORITY OF 6800 INSTRUCTIONS

BINARY OPERATORS

- BINARY OPERATIONS (b) REQUIRE TWO SOURCE OPERANDS
- DESTINATION MUST BE SPECIFIED; IMPLICIT OR EXPLICIT.
- $(R) \leftarrow (R)\ b\ (n)$; $(R) \leftarrow (R)\ b\ M(n)$;
  $(R) \leftarrow (R)\ b\ M(IX+n)$; $(R1) \leftarrow (R1)\ b\ (R2)$
- NOTE : DESTINATION ALWAYS A REGISTER (A or B)

• ARITHMETIC (add + subtract)	ADD	SUB
	ABA	SBA
	ADC	SBC
• LOGICAL (AND, OR, EXCLUSIVE-OR, BIT TEST, COMPARE)	AND	BIT (R.M)
	EOR	CMP (R-M)
	ORA	CBA (A-B)

8-BIT UNARY OPERATORS

- UNARY OPERATORS REQUIRE A SINGLE OPERAND
- SOURCE AND DESTINATION ARE THE SAME
- $M(n) \leftarrow M(n)\ u$; $M(IX+n) \leftarrow M(IX+n)\ u$
  $(R) \leftarrow (R)\ u$ NB R IS A OR B

CLR	$\leftarrow 00$
COM	$\leftarrow \overline{X}$
NEG	$\leftarrow (00-X)$
DEC	$\leftarrow (X-1)$
INC	$\leftarrow (X+1)$
TST	$\leftarrow (X-00)$

- RANGE OF SHIFT INSTRUCTIONS

1. LOGICAL SHIFT (LEFT/RIGHT)
2. ARITHMETIC SHIFT (LEFT/RIGHT)
3. ROTATE (LEFT RIGHT)

- OPERATE ON A, B OR M( )

16 BIT OPERATORS

- BINARY AND UNARY OPERATORS
- OPERATE ON SP AND IX

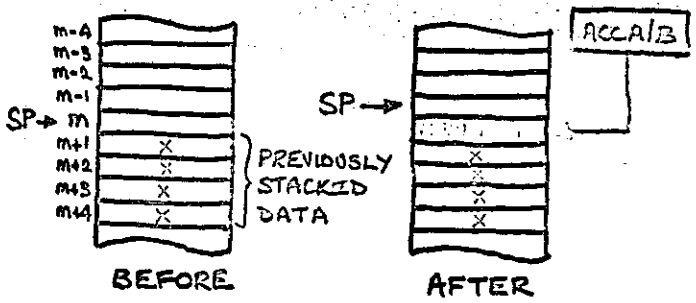
CPX	(IX) − (M,n)
TXS	(SP) ← (IX−1)
TSX	(IX) ← (SP+1)
DEX	(IX) ← (IX)−1
DES	(SP) ← (SP)−1
INX	(IX) ← (IX)+1
INS	(SP) ← (SP)+1

- STACK OPERATORS
  - 'stack' is a sequential list of data stored in RWM.
  - SP access the data on the stack
  - stack operates on a last-in/first-out basis
  - used for 'data handling'
  - STACK OPERATIONS

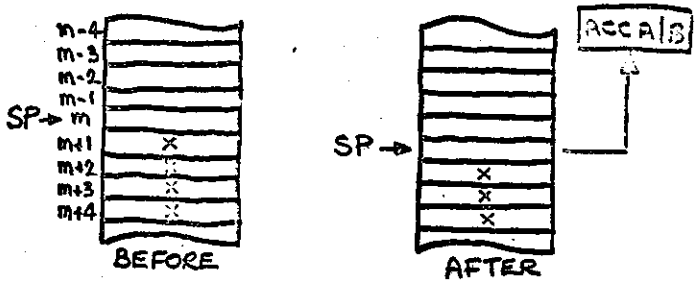
    PSH (A,B) : SP(M) ← (A,B); (SP) ← (SP)-1

    PUL (A,B) : (SP) ← (SP)+1 ; (A,B) ← SP(M)

PUSH (PSH)



PULL (PUL)



CONDITION FLAGS

- 6800 CONTAINS A 6-BIT CCR WHOSE CONTENTS ARE ALTERED BY EXECUTING ONE OF THE PREDEFINED OPERATORS.
- REFER TO 6800 IS FOR DETAILS
- CONTENTS OF CCR :

5	4	3	2	1	0
H	I	N	Z	V	C

H = half carry flag
N = negative flag
  N=1 if bit 7 of result = 1
  else N=0

I = interrupt mask
Z = zero flag
  Z=1 if result = 0
  else Z=0

V = overflow flag
  V=1 if result produced a 2's compl. overflow
  else V=0

C = carry flag
  C=1 if carry out of bit 7 of the result of an arithmetic operation
  else C=0

- CCR may be manipulated by a set of instructions

CLC	(C) ← 0	SEI	(I) ← 1
CLI	(I) ← 0	SEV	(V) ← 1
CLV	(V) ← 0	TAP	(CCR) ← (A)
SEC	(C) ← 1	TPA	(A) ← (CCR)

- CONDITION FLAGS ARE USED EXTENSIVELY IN PROGRAM CONTROL INSTRUCTIONS

PROGRAM CONTROL

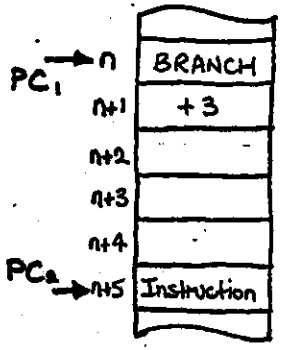
- VARIETY OF PC MECHANISMS ARE NEEDED
- NORMAL CONTROL FLOW IS SEQUENTIAL, NECESSARY TO ALTER THIS AND 'JUMP' TO SOME OTHER PART OF THE TOTAL PROGRAM.
- IF A SECTION OF PROGRAM IS REQUIRED MANY TIMES IN DIFFERENT PARTS OF THE TOTAL PROGRAM, IN ORDER TO AVOID REPLICATION OF THE CODE A SPECIAL TYPE OF JUMP MECHANISM IS REQUIRED (SUBROUTINE CALL) TO A SINGLE CODE SEGMENT, WHICH MAY BE ACCESSED MANY TIMES.

- 6800 PROVIDES 3 TYPES OF PROGRAM CONTROL INSTRUCTION
  - BRANCH
  - JUMP
  - SUBROUTINE ACCESS

- IN PROGRAM CONTROL INSTRUCTIONS ONLY THE PROGRAM COUNTER IS ALTERED.

BRANCH INSTRUCTIONS

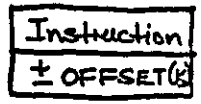
- PROGRAM CONTROL FLOW IS ALTERED TO A POINT RELATIVE TO THE CURRENT PROGRAM COUNTER.



AFTER EXECUTION OF THE 'BRANCH' INSTRUCTION AT PC_1, THE PC IS ALTERED TO PC_2 where

$PC_2 = (PC_1 + 2) + 3$

- GENERAL FORM



- THE OFFSET MAY BE POSITIVE OR NEGATIVE (bit M = sign (0 = '+', 1 = '−').
- MAGNITUDE OF THE OFFSET = 127
- RANGE OF BRANCH INSTRUCTIONS

$(PC+2) - 127 \leq DA \leq (PC+2) + 127$

where DA = destination address.

- CONDITIONAL + UNCONDITIONAL BRANCHES

- UNCONDITIONAL BRANCH
  - PC IS ALWAYS ALTERED
- CONDITIONAL BRANCH
  - PC IS ALTERED IF A CERTAIN CONDITION IS MET, OTHERWISE THE NORMAL CONTROL FLOW CONTINUES.

IF condition = 1 THEN (PC) := (PC)+2±K
ELSE (PC) := (PC)+2

BRA : BRANCH ALWAYS

BCC	C=0	BLT	N⊕V=1
BCS	C=1	BLS	C+Z=1
BEQ	Z=1	BMI	N=1
BGE	N⊕V=0	BNE	Z=0
BGT	Z+(N⊕V)=0	BVC	V=0
BHI	C+Z=0	BVS	V=1
BLE	Z+(N⊕V)=1	BPL	N=0

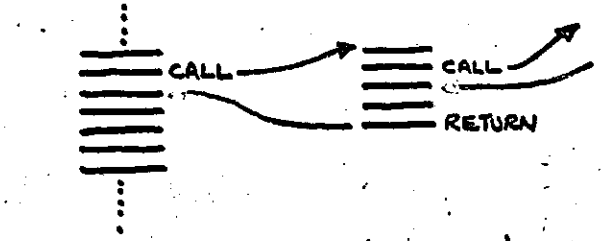
JUMP INSTRUCTION

- PROGRAM FLOW IS ALTERED TO ANYWHERE IN MEMORY.

JMP (PC) ← (IX)+K (indexed)
(PC) ← K (extended)

SUBROUTINES

- SUBROUTINES AVOID REPLICATION OF CODE
- WHEN ACCESSING SUBROUTINES THE 6800 STORES AWAY THE ADDRESS OF THE INSTRUCTION FOLLOWING THE CALL SO THAT WHEN THE SUBROUTINE ACTIONS HAVE BEEN COMPLETED, CONTROL IS RETURNED TO THE MAIN PROGRAM
- THE PC IS STORED ON THE STACK



- SUBROUTINES MAY BE 'NESTED' TO ANY LEVEL. THE ONLY LIMITATION IS THE SIZE OF THE STACK IN RWM.
- OPERATION IS SIMILAR TO THE PSH & PUL INSTRUCTIONS { PSH – CALL, PUL – RETURN }

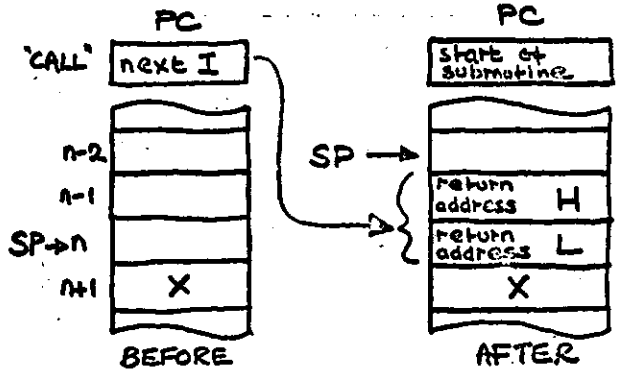
SUBROUTINE ACCESS INSTRUCTIONS

BSR SP(M) ← (PC)+2 ; (SP) ← (SP)-2

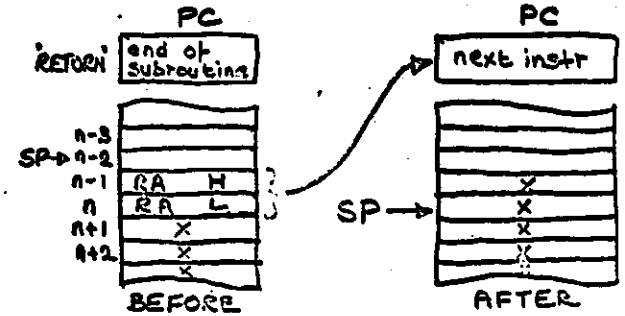
JSR SP(M) ← (PC)+2 ; (SP) ← (SP)-2

RTS (PC) ← SP(M) ; (SP) ← (SP)+2

- CALL (BSR, JSR)



- RETURN (RTS)



INPUT-OUTPUT

- THE 6800 DOES NOT HAVE ANY SPECIFIC I/O INSTRUCTIONS
- MEMORY MAPPED I/O IS USED

INTERRUPTS

- SOFTWARE, EXTERNAL, NON-MASKABLE
- WHEN AN INTERRUPT IS OBEYED THE REGISTERS ARE AUTOMATICALLY 'STACKED' AND THE PC IS FORCED TO A SPECIFIC VALUE
- KNOWN AS 'INTERRUPT VECTORS' (IV)
- THE PC WILL BE ALTERED TO THE CONTENTS OF THE INTERRUPT VECTOR
  - 'SOFTWARE' IV $FFFA, FFFB
    'INTERRUPT RQ' IV $FFF8, FFF9
    'NON-MASKABLE' IV $FFFC, FFFD
- WHEN THE INTERRUPT HAS BEEN 'SERVICED', THE REGISTERS ARE UNSTACKED AND CONTROL IS RETURNED TO THE 'MAIN PROGRAM'

INTERRUPTS ARE EXTERNAL, ASYNCHRONOUS EVENTS WHICH REQUIRE IMMEDIATE ACTION

SUMMARY

- EXAMINED THE REQUIREMENTS FOR AN INSTRUCTION SET
- LOOKED AT THE FACILITIES PROVIDED BY 6800
  - TYPICAL OF 8-bit μP's.
  - TAILORED TO THE ARCHITECTURE

- LEARN THE INSTRUCTION SET OF ONE MICROPROCESSOR
- WRITE PROGRAMS FOR THE μP
  - SEE WHAT YOU CAN DO
  - SEE WHAT YOU CANNOT DO
- EXAMINE THE INSTRUCTION SETS OF OTHER MICROPROCESSORS

GENERAL RULE

GET TO KNOW ONE MICROPROCESSOR WELL AND STICK TO IT UNTIL YOU ARE EXPERIENCED