How assembler works

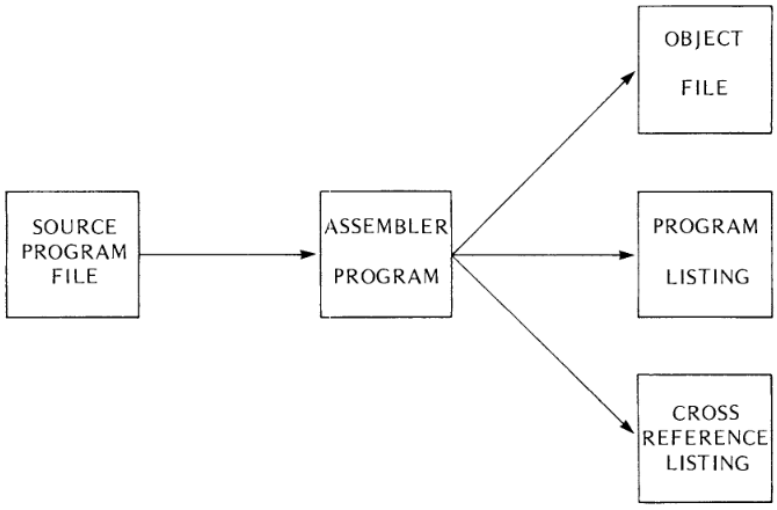


Figure 1-1. Assembler Outputs

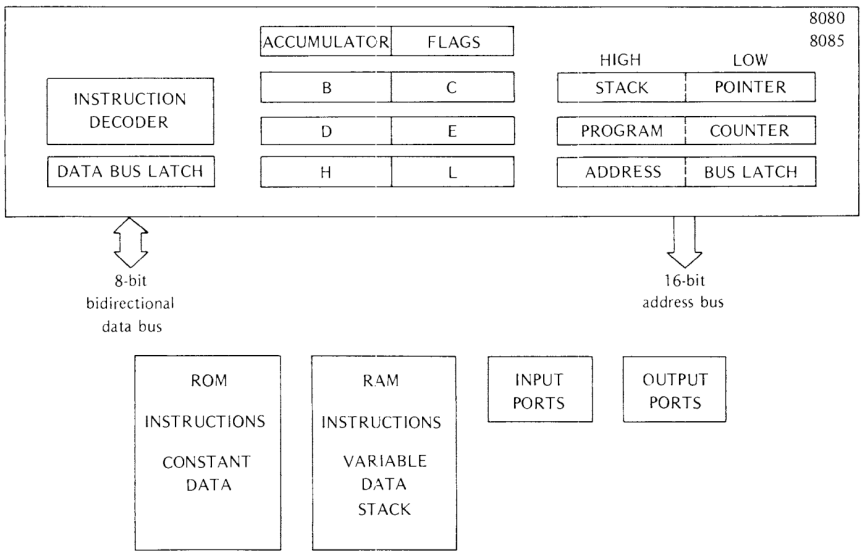
Figure 1: assembler_output

- object file Object file is what is loaded to ROM. 8085 is 40 pinned because the intellec developement system only supported 40 pins.
- program listing record for source program and object code. Think of it as table
- cross reference listing another record that is one of the diagnostic tools provided by the assembler.

8085

- memory
- Program counter
- Work registers
- Condition flags
- Stack and stack pointer
- IO ports
- instruction set

Registers 8085



Internal registers of 8085

sybmolic name of registers

Condition flags Carry , sign , zero , parity and Auxillary Carry.

Addressing Modes

- *Implied Addressing* : STC, DAA instructions are implied by the instruction function
- *Register Addressing*: CMP E. specify one regiser, another one is by default accumulator so it deals with 8-bit values. Except some like `PCHL` which exchanges the contents of program counter with H and L register.
- *Immediate Addressing*: MVI D, OFFH, LXI SP,30FFH (16bit load). Specify register , then what to load.
- *Direct Addressing*: JMP 1000H

- *Register Indirect addressing* : references memory by a register pair. MOV M,C moves the contents of the C register to memory address stored in the H and L pair.

MOV	Move
MVI	Move Immediate
LDA	Load Accumulator Directly from Memory
STA	Store Accumulator Directly in Memory
LHLD	Load H and L Registers Directly from Mem
SHLD	Store H and L Registers Directly in Memory

Instruction Naming conventions

An 'X' in the name of a data transfer instruction implies that it deals with a register pair:

LXI	Load Register Pair with Immediate data
LDAX	Load Accumulator from Address in Register Pair
STAX	Store Accumulator in Address in Register Pair
XCHG	Exchange H and L with D and E
XTHL	Exchange Top of Stack with H and L

Arithmetic Group. The arithmetic instructions add, subtract, increment, or decrement data in registers or memory.

ADD	Add to Accumulator
ADI	Add Immediate Data to Accumulator
ADC	Add to Accumulator Using Carry Flag
ACI	Add Immediate Data to Accumulator Using Carry Flag
SUB	Subtract from Accumulator
SUI	Subtract Immediate Data from Accumulator
SBB	Subtract from Accumulator Using Borrow (Carry) Flag
SBI	Subtract Immediate from Accumulator Using Borrow
INR	Increment Specified Byte by One
DCR	Decrement Specified Byte by One
INX	Increment Register Pair by One
DCX	Decrement Register Pair by One
DAD	Double Register Add: Add Contents of Register Pair to H and L Register Pair

Thus, the conditional branching instructions are specified as follows:

Jumps	*Calls*	*Returns*	
JC	CC	RC	(Carry)
JNC	CNC	RNC	(No Carry)
JZ	CZ	RZ	(Zero)
JNZ	CNZ	RNZ	(Not Zero)
JP	CP	RP	(Plus)
JM	CM	RM	(Minus)
JPE	CPE	RPE	(Parity Even)
JPO	CPO	RPO	(Parity Odd)

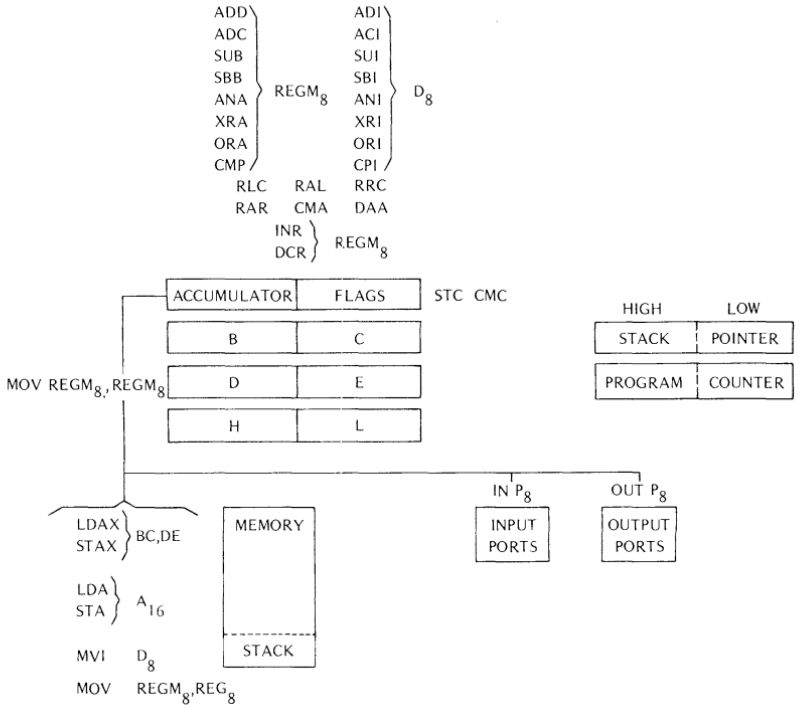


Figure 2: hardware_instruction_summary

Concepts of Assembly Language

- character set - Delimiters: characters with special meaning to the assembler. Delimiters define the end of source statement, a field or component.

Operand field information operand field information

Two's complement representation of data

Symbolic Addressing