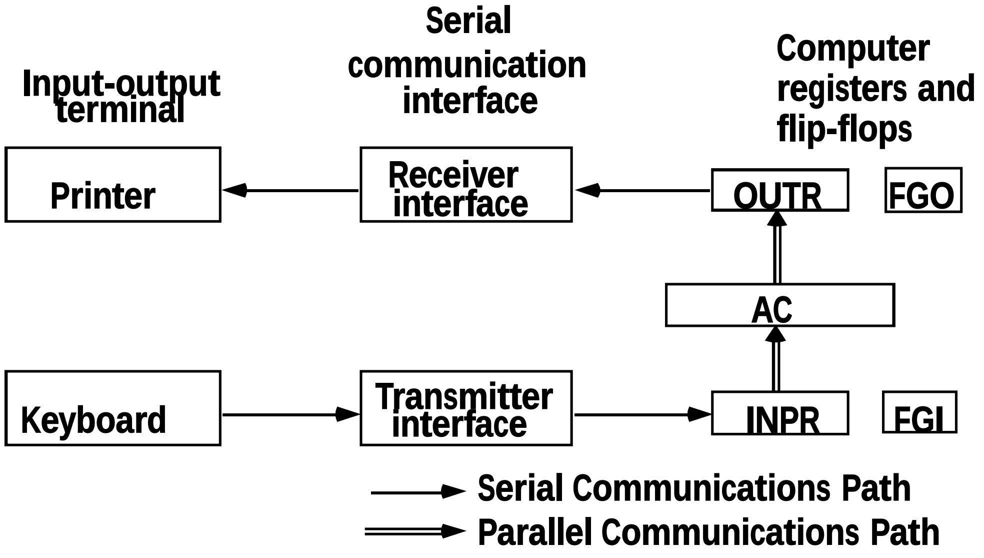
Input-Output and Interrupt

Input-Output Configuration

- A Terminal with a keyboard and a Printer

INPR	Input register - 8 bits
OUTR	Output register - 8 bits
FGI	Input flag - 1 bit
FGO	Output flag - 1 bit
IEN	Interrupt enable - 1 bit


Input-output terminal
Serial communication interface
Computer registers and flip-flops
Printer
Receiver interface
OUTR
FGO
AC
Keyboard
Transmitter interface
INPR
FGI
Serial Communications Path
Parallel Communications Path

Input-Output Configuration

- Input and output terminal sends and receives serial information's
- Each quantity of information has 8 bits of alphanumeric code.
- The serial information from the keyboard is shifted into I/P register INPR.
- The serial information for the printer is stored in the O/P register OUTR.
- These two registers communicate with a communication interface serially and with AC in parallel
- Transmitter interface receives serial information from keyboard and transmit to INPR.
- Receiver interface receives information from OUTR and sends to printer serially.

Input Register

- Input register INPR consist of 8 bits and holds alphanumeric input information.
- The 1 bit input flag FGI is a control flip-flop which is set to 1 when new information is available in INPR and it is cleared to 0 when the information is accepted by the computer.
- The process of information transfer is as follows:

  1. Initially the input flag FGI is cleared to 0.
  2. When a key is pressed in the keyboard, an 8 bit alphanumeric code is shifted to INPR and the FGI is set to 1.
  3. The computer checks the flag bit, if it is 1 the information from INPR is transferred in parallel into AC and FGI is cleared to 0.
  4. Once the FGI is cleared, new information can be shifted into INPR by striking another key.

Output Register

- The output register OUTR works similarly but direction of information flow is reversed.
- The process of information transfer is as follows:

1. Initially the output flag FGO is set to 1.
2. The computer checks the flag bit, if it is 1 the information from AC is transferred in parallel into OUTR and FGO is cleared to 0.
3. The output device accepts the coded information , prints the corresponding character and when the operation is completed , it sets FGO to 1
4. The computer does not load a new character into OUTR when FGO is 0 because this indicates that output device is in the process of printing the character.

INPUT-OUTPUT INSTRUCTIONS

- I/p and O/p instruction have the opcode 1111 and recognized by the control when D7=1 and I=1.
- The remaining bits of instruction specify the particular operation.
- The control function and micro operation for input output instruction are listed below

INPUT-OUTPUT INSTRUCTIONS

$D_7IT_3 = p$
$IR(i) = B_i$, i = 6, …, 11

p:		SC ← 0	Clear SC
INP	pB_{11}:	AC(0-7) ← INPR, FGI ← 0	Input char. to AC
OUT	pB_{10}:	OUTR ← AC(0-7), FGO ← 0	Output char from AC
SKI	pB_9:	if(FGI = 1) then (PC ← PC + 1)	Skip on input flag
SKO	pB_8:	if(FGO = 1) then (PC ← PC + 1)	Skip on output flag
ION	pB_7:	IEN ← 1	Interrupt enable on
IOF	pB_6:	IEN ← 0	Interrupt enable off

BASIC COMPUTER INSTRUCTIONS

	Hex Code		
Symbol	*I = 0*	*I = 1*	*Description*
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instr. if AC is positive
SNA	7008		Skip next instr. if AC is negative
SZA	7004		Skip next instr. if AC is zero
SZE	7002		Skip next instr. if E is zero
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off