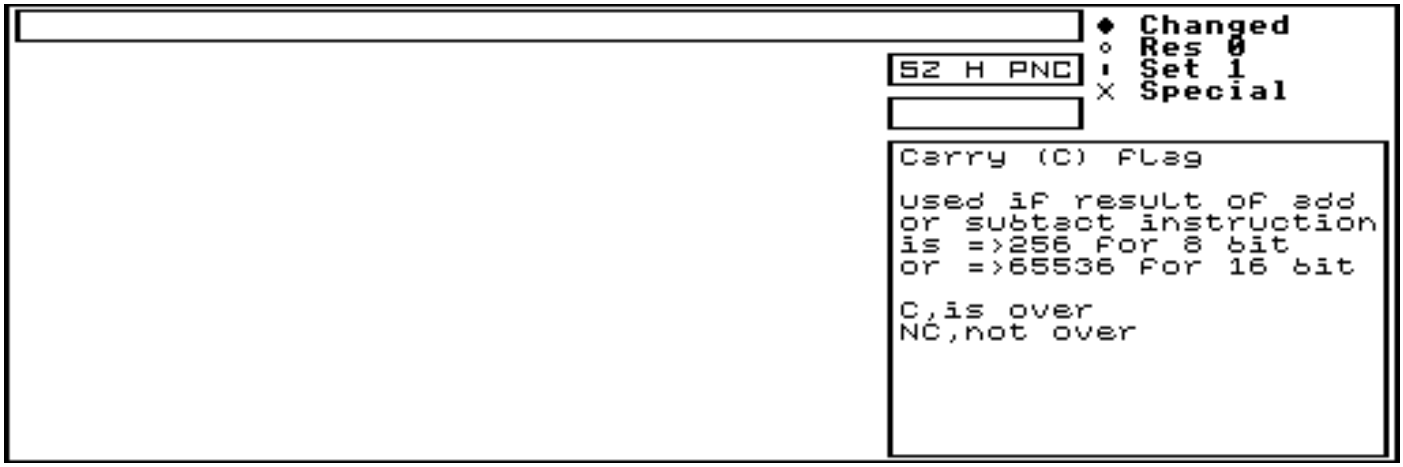
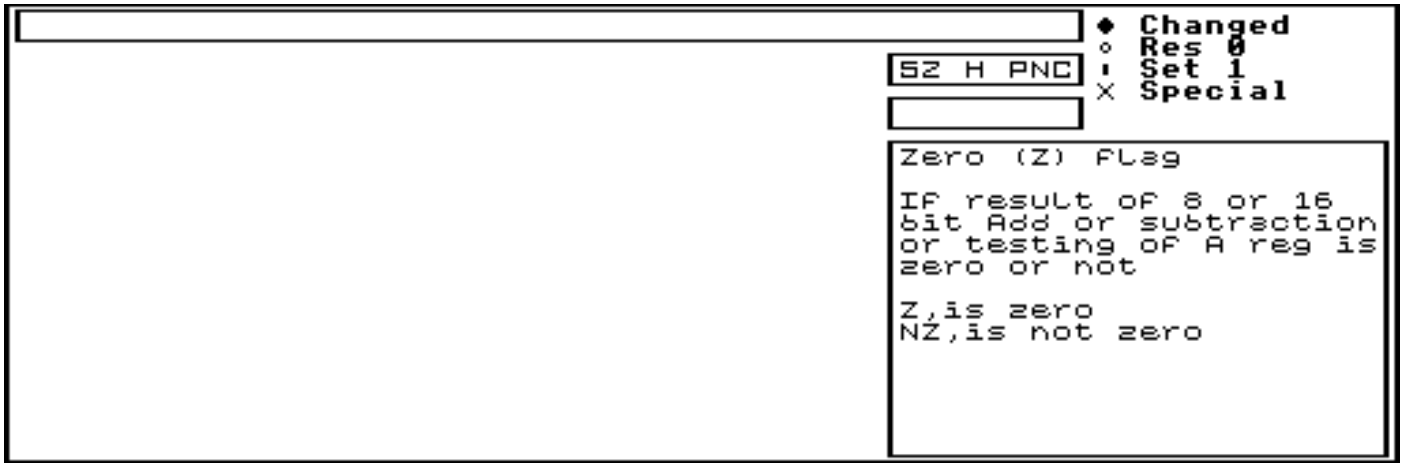
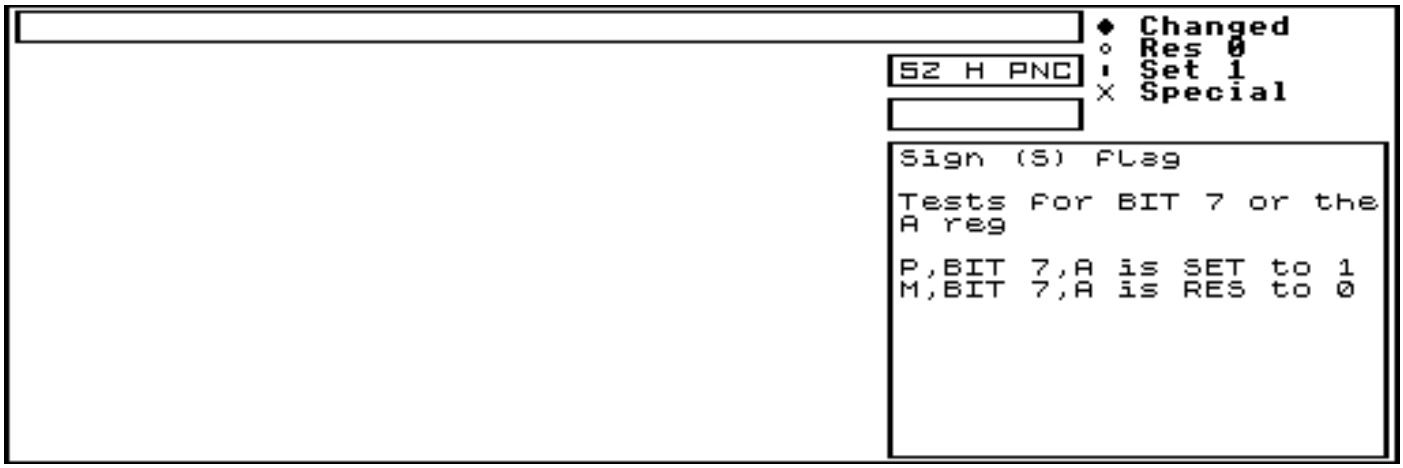
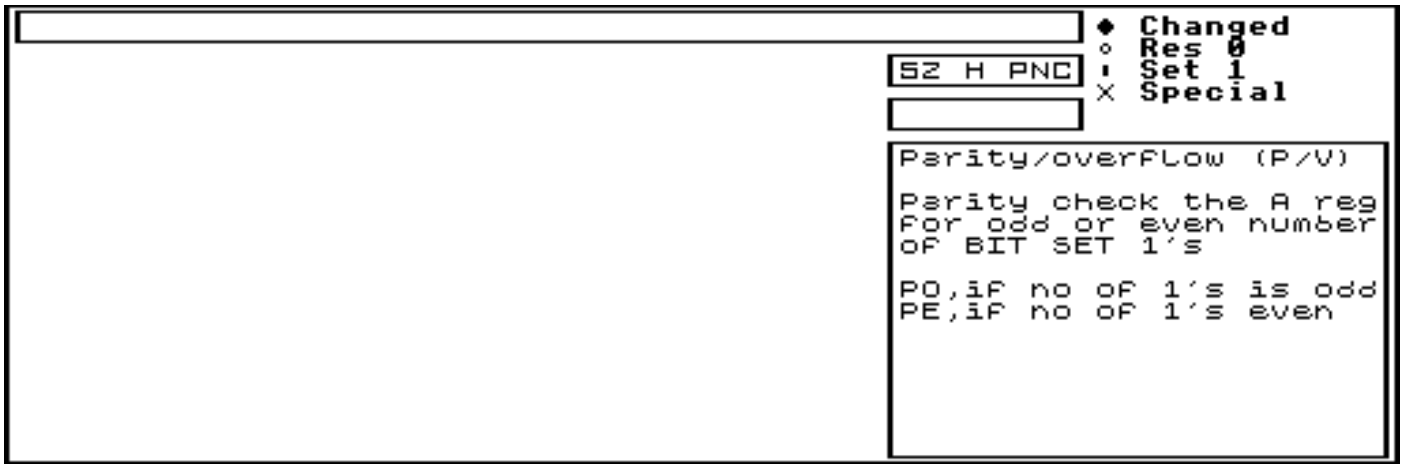

◆ Changed
◦ Res 0
! Set 1
X Special
SZ H PNC
Carry (C) Flag
used if result of add or subtact instruction is =>256 for 8 bit or =>65536 for 16 bit
C,is over
NC,not over


◆ Changed
◦ Res 0
! Set 1
X Special
SZ H PNC
Zero (Z) Flag
IF result of 8 or 16 bit Add or subtraction or testing of A reg is zero or not
Z,is zero
NZ,is not zero


◆ Changed
◦ Res 0
! Set 1
X Special
SZ H PNC
Sign (S) Flag
Tests for BIT 7 or the A reg
P,BIT 7,A is SET to 1
M,BIT 7,A is RES to 0


◆ Changed
◦ Res 0
! Set 1
X Special
SZ H PNC
Parity/overFlow (P/V)
Parity check the A reg for odd or even number of BIT SET 1's
PO,if no of 1's is odd
PE,if no of 1's even

ADC A,s

(HL)	142	8E	7
A	143	8F	4
B	136	88	4
C	137	89	4
D	138	8A	4
E	139	8B	4
H	140	8C	4
L	141	8D	4
n	206,n	CE n	7
IXh	221,140	DD 8C	8
IYh	253,140	FD 8C	8
IXl	221,141	DD 8D	8
IYl	253,140	FD 8D	8
(IX+d)	221,142,d	DD 8E d	19
(IY+d)	253,142,d	FD 8E d	19

SZ H PNC
◆◆ ◆ ◆○◆

◆ Changed
○ Res 0
▮ Set 1
X Special

```
A= A+reg+carry flag

SCF will SET carry to1
SUB A,A will RES carry
LD   B,10   ;B=10
LD   A,50   ;A=50
ADC A,B     ;A=A+B+carry
JR   C,over;IF A=>256
```

ADC HL,s

BC	237,74	ED 4A	15
DE	237,90	ED 5A	15
HL	237,106	ED 6A	15
SP	237,122	ED 7A	15

SZ H PNC
◆◆ ◆○◆

◆ Changed
○ Res 0
▮ Set 1
X Special

```
HL= HL+reg+carry flag

H flag set if there is
  a carry from bit 11
```

ADD A,s

(HL)	134	86	7
A	135	87	4
B	128	80	4
C	129	81	4
D	130	82	4
E	131	83	4
H	132	84	4
L	133	85	4
n	198,n	C6 n	7
IXh	221,132	DD 84	8
IYh	253,132	FD 84	8
IXl	221,133	DD 85	8
IYl	253,133	FD 85	8
(IX+d)	221,134,d	DD 86 d	19
(IY+d)	253,134,d	FD 86 d	19

SZ H PNC
◆◆ ◆ ◆○◆

◆ Changed
○ Res 0
▮ Set 1
X Special

```
A= A+reg

Adding two registers
with HL=result (0-510)

    LD   H,0    ;H=0
    LD   A,100  ;A=100
    LD   C,200  ;C=200
    ADD  A,C    ;A=A+C
    JR   C,ov   ;A=>256
    LD   L,A    ;L=A
    RET
ov:INC  H       ;H=H+1
    RET
```

ADD HL,s

BC	9	09	11
DE	25	19	11
HL	41	29	11
SP	57	39	11

SZ H PNC
○◆

◆ Changed
○ Res 0
▮ Set 1
X Special

```
HL= HL+reg

C is SET by carry from
  BIT 15,otherwise RES

H is SET by carry from
  BIT 11

LD   HL,10000 ;HL=10000
LD   DE,20000 ;DE=20000
ADD  HL,DE    ;HL=HL+DE
JR   C,over   ;HL is
                >=65536
```

ADD IX,s / ADD IY,s

IX,BC	221,9	DD	09	15
IX,DE	221,25	DD	19	15
IX,IX	221,41	DD	29	15
IX,SP	221,57	DD	39	15
IY,BC	253,9	FD	09	15
IY,DE	253,25	FD	19	15
IY,IY	253,41	FD	29	15
IY,SP	253,57	FD	39	15

SZ H PNC
○◆

◆ Changed
○ Res 0
▮ Set 1
X Special

```
IX=IX+reg    or
IY=IY+reg

C is SET by carry from
  BIT 15,otherwise RES

H is SET by carry from
  BIT 11
```

AND s

(HL)	166	A6		7
A	167	A7		4
B	160	A0		4
C	161	A1		4
D	162	A2		4
E	163	A3		4
H	164	A4		4
L	165	A5		4
n	230,n	E6	n	7
IXh	221,164	DD	A4	8
IYh	253,164	FD	A4	8
IXl	221,165	DD	A5	8
IYl	253,165	FD	A5	8
(IX+d)	221,166,d	DD	A6 d	19
(IY+d)	253,166,d	FD	A6 d	19

SZ H PNC
◆◆ ▮ ◆○○

◆ Changed
○ Res 0
▮ Set 1
X Special

```
A= A and reg

LD  A,%00011010
LD  B,%11001011
AND B
A now= 00001010
Each bit is compared
by both regs and put
in the A reg, bit SET
to 1 only if there are
two 1's, others to 0's
AND %00001111 will RES
bits 7,6,5,4 and leave
bits 3,2,1,0 alone.
```

BIT b,s (b is number from 0-7)

b,(HL)	203,70+(b*8)	CB	46+(b*08	12
b,A	203,71+(b*8)	CB	47+(b*08	8
b,B	203,64+(b*8)	CB	40+(b+08	8
b,C	203,65+(b*8)	CB	41+(b+08	8
b,D	203,66+(b*8)	CB	42+(b*08	8
b,E	203,67+(b*8)	CB	43+(b*08	8
b,H	203,68+(b*8)	CB	44+(b*08	8
b,L	203,69+(b*8)	CB	45+(b*08	8
b,(IXd	221,203,d,70+(b*8	DD	CB 46+(b*08	20
b,(IYd	253,203 d,70+(b*8	FD	CB 46+(b*08	20

SZ H PNC
◆◆ ▮ ◆○○

◆ Changed
○ Res 0
▮ Set 1
X Special

```
Test bit b of register
Z flag =0 if BIT RES 0
Z flag =1 if BIT SET 1

BIT 0,B ;test if BIT 0
        of B reg= 0or1
JR  Z,zero ;if BIT 0=0


BIT 6,L ;test if BIT 6
        of L reg= 0or1
JR  NZ,one ;if BIT 6=1
```

CALL s / CALL flag,s (flag C,M,NC,NZ,P,PE,PO,Z)

nn	205,n,n	CD	n n		17
C,nn	220,n,n	DC	n n	17	10
M,nn	252,n,n	FC	n n	17	10
NC,nn	212,n,n	D4	n n	17	10
NZ,nn	196,n,n	C4	n n	17	10
P,nn	244,n,n	F4	n n	17	10
PE,nn	236,n,n	EC	n n	17	10
PO,nn	228,n,n	E4	n n	17	10
Z,nn	204,n,n	CC	n n	17	10

SZ H PNC

◆ Changed
○ Res 0
▮ Set 1
X Special

```
CALL  address  or
CALL  flag status,addr

CALL 32768;GOSUB 32768

CALL Z,40000 ;GOSUB
40000 only if Z flag=0

CALL is like Sam's
GOSUB,RET is like Sams
RETURN,The SP stack is
used like Sam's GOSUB
stack. POP reg is like
Sam's  POP variable.
```

CCF

CCF	63	3F	4	

SZ H PNC
°◆

◆ Changed
° Res 0
' Set 1
X Special

```
Carry flag toggle

If Carry is SET to 1
CCF will RES 0 carry

If Carry is RES to 0
CCF will SET 1 carry
```

CP s

(HL)	190	BE	7	
A	191	BF	4	
B	184	B8	4	
C	185	B9	4	
D	186	BA	4	
E	187	BB	4	
H	188	BC	4	
L	189	BD	4	
n	254,n	FE n	7	
IXh	221,188	DD BC	8	
IYh	253,188	FD BC	8	
IXl	221,189	DD BD	8	
IYl	253,189	FD BD	8	
(IX+d)	221,190,d	DD BE d	19	
(IY+d)	253,190,d	FD BE d	19	

SZ H PNC
◆◆ ◆ ◆'◆

◆ Changed
° Res 0
' Set 1
X Special

```
reg compared with the
A reg,result in flags

LD B,20 ;A=20
LD A,10 ;B=10
CP B    ;compare A & B
JR Z,equal ;B=A
JR NC,over ;B>A
JR C,less  ;B=<A
```

CPD

CPD	237,169	ED A9	16	

SZ H PNC
◆X ◆ X'

◆ Changed
° Res 0
' Set 1
X Special

```
CPD is the equivelent
    of executing the 3
    following opcodes.
CP (HL)
DEC HL
DEC BC

Z flag=1 if A=(HL)
P flag=0 if BC=0
```

CPDR

CPDR	237,185	ED B9	16 21	

SZ H PNC
◆X ◆ X'

◆ Changed
° Res 0
' Set 1
X Special

```
CPDR is the equivelent
    of executing the 4
    following opcodes.
Loop: CP (HL)
      DEC HL
      DEC BC
      JP  Loop

returns if :-
A=(HL);Z flag would =1
BC=0  ;P flag would =0

CPDR   mostly used for
Alphabetical sorting
```

CPI

CPI	237,161	ED A1	16	SZ H PNC	◆ Changed
				◆X ◆ X▮	◦ Res 0
					▮ Set 1
					X Special

```
CPI  is the equivelent
     of executing the 3
     following opcodes.
CP (HL)
INC HL
DEC BC

Z flag=1 if A=(HL)
P flag=0 if BC=0
```

CPIR

CPIR	237,177	ED B1	16 21	SZ H PNC	◆ Changed
				◆X ◆ X▮	◦ Res 0
					▮ Set 1
					X Special

```
CPIR is the equivelent
     of executing the 4
     following opcodes.
Loop:  CP (HL)
       INC HL
       DEC BC
       JP  Loop

returns if :-
A=(HL);Z flag would =1
BC=0  ;P flag would =0

CPIR  is excellent for
searching memory fast
```

CPL

CPL	47	2F	4	SZ H PNC	◆ Changed
				▮ ▮	◦ Res 0
					▮ Set 1
					X Special

```
A=inverse A reg

LD  A,%00101101
CPL

A now= 11010010

Each bit in A reg is
inversed,so if BIT 6
=1 then BIT 0 now =0
if BIT 3=0 then BIT 3
now =1
```

DAA

DAA	39	27	4	SZ H PNC	◆ Changed
				◆◆ ◆ ◆ ◆	◦ Res 0
					▮ Set 1
					X Special

```
Never used in Normal
programming of Games
Software,Utilies etc.

Sometimes used in
Scientific Mathmatics
It's basically used as
part of a way to add
two seperate numbers
from (0-99), held in a
special form in the A
register.
You are unlikely to
use this opcode.
```

DEC s

(HL)	53	35	11	SZ H PNC
A	61	3D	4	
B	5	05	4	◆◆ ◆ ◆!
C	13	0D	4	
D	21	15	4	
E	29	1D	4	
H	37	25	4	
L	45	2D	4	
IXh	221,37	DD 25	8	
IYh	253,37	FD 25	8	
IXl	221,45	DD 2D	8	
IYl	253,45	FD 2D	8	

◆ Changed
◦ Res 0
! Set 1
X Special

```
operand=operand-1

LD    C,10 ;C=10
DEC   C    ;C=C-1
JR    Z,ze ;iF C=0
```

DEC ss

BC	11	0B	6	SZ H PNC
DE	27	1B	6	
HL	43	43	6	
SP	59	3B	6	
IX	221,43	DD 2B	10	
IY	253,43	FD 2B	10	

◆ Changed
◦ Res 0
! Set 1
X Special

```
operand=operand-1

LD  HL,32768 ;HL=32768
DEC HL       ;HL=HL-1
LD  A,H      ;test iF
OR  L        ;HL=0
JR  Z,zero   ;iF so JR
               to zero
```

DI

DI	243	F3	4	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

turn on (Disable) Maskable Interrupt Sam's Rom keyscanning is turned off etc to allow Machine code to Run a bit faster.

EI (Enable Interrupts) to turn DI off so that keyscanning etc are on again, Never return to Sam basic with DI on, otherwise keyboard will Lock up

DJNZ ofset

label	16,d	10 d	8 13	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
      LD   B,100
Loop: INC  HL
      LD   A,(HL)
      CP   13
      RET  Z
      DJNZ Loop
      RET
```

DJNZ will DEC B reg IF B=0 cont prog,ELSE goto the address label in an ofset form, From address DJNZ as upto 126 bytes backwards or upto 129 bytes forward

EI

EI	251	FB	4	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

EI (Enable Maskable Interupts)

IF you used DI to stop the Rom keyscanning etc use EI to return back Sam to normal Interupt mode.

EX AF,AF'

	8	08	4	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

◆◆◆◆◆◆◆◆

Exchange / Swap

The A reg and flags with ALterative A reg and flags

EX DE,HL

	235	EB	4	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

Exchange / Swap

The HL register with the DE register

EX (SP),HL

	227	E3	19	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

Exchange / Swap

Swap what is on the stack with HL reg

EX (SP),IX / EX (SP),IY

◆ Changed
◦ Res 0
¦ Set 1
X Special

(SP)IX	221,227	DD E3	23	SZ H PNC	
(SP)IY	253,227	FD E3	23		

Exchange / Swap

Swap what is on the stack with IX or IY reg

EXX

◆ Changed
◦ Res 0
¦ Set 1
X Special

EXX	217	D9	4	SZ H PNC	

Exchange / Swap

Normal to Alternative

HL to HL' and
DE to DE' vice
BC to BC' versa

HALT

◆ Changed
◦ Res 0
¦ Set 1
X Special

HALT	118	76	4	SZ H PNC	

Sam halts what it is doing (just runs NOPS) until an interrupt or reset is detected, at this point the TV is just about to start displaying the next TV Frame Picture this happens evert 1/50 sec

Games programmers use this HALT instruction to know when to update the TV picture.

IM 0

◆ Changed
◦ Res 0
¦ Set 1
X Special

IM 0	237,70	ED 46	8	SZ H PNC	

This is no use to the programmer, is used by Sam as the main interrupt,it cannot be stopped by DI etc

IM 1

					SZ H PNC
IM 1	237,86	ED 56	8		

◆ Changed
◦ Res 0
▮ Set 1
X Special

When an interrupt occurs every 1/50 of a sec the Machine code that is being run is temporary halted and Sam goes on to an interrupt handler at #0038 or 56 decimal to read the keyboard, mouse,Line interrupt etc,when done the your Machine program you were running continues from where you left of

IM 2

					SZ H PNC
IM 2	237,94	ED 5E	8		

◆ Changed
◦ Res 0
▮ Set 1
X Special

The most powerful and useful interrupt for programmers,it enables you to create your own interrupt routine in memory and force this interupt mode 2 to any address in the High bank memory area. You can turn or or off the interrupt easily you don't have to use DI or EI.

IN r,(C)

					SZ H PNC
A,(C)	237,120	ED 78	12		
B,(C)	237,64	ED 40	12		
C,(C)	237,72	ED 48	12		◆◆ ◆ ◆◦
D,(C)	237,80	ED 50	12		
E,(C)	237,88	ED 58	12		
H,(C)	237,96	ED 60	12		
L,(C)	237,140	ED 68	12		

◆ Changed
◦ Res 0
▮ Set 1
X Special

```
register=PEEK port C


LD C,254 ;C the port
IN A,(C) ;A=PEEK portC
```

Port 254 can be read for keys from the key board, even if the DI is on. Some ports are double need to LD BC, with 16 bit port then IN A,(C).

IN A,(n)

					SZ H PNC
A,(n)	219,n	DB n	11		

◆ Changed
◦ Res 0
▮ Set 1
X Special

```
A=PEEK port n

IN A,(252) ;A=PEEK
             Port 251
```

This will read which memory pages are used for The screen display

INC s

(HL)	52	34	11
A	60	3C	4
B	4	04	4
C	12	0C	4
D	20	14	4
E	29	1C	4
H	36	24	4
L	44	2C	4
IXh	221,36	DD 24	8
IYh	253,36	FD 24	8
IXl	221,44	DD 2C	8
IYl	253,44	FD 2C	8
(IX+d)	221,52,d	DD 34 d	23
(IY+d)	253,52,d	FD 34 d	23

SZ H PNC
◆◆ ◆ ◆○

◆ Changed
○ Res 0
▮ Set 1
X Special

```
operand=operand+1

LD   A,10 ;A=10
INC  A    ;A=A+1
JR   C,ov ;iF c>=256
```

INC ss

BC	3	03	6
DE	19	13	6
HL	35	23	6
SP	51	33	6
IX	221,35	DD 23	10
IY	253,35	FD 23	10

SZ H PNC

◆ Changed
○ Res 0
▮ Set 1
X Special

```
operand=operand+1

LD   HL,32768 ;HL=32768
INC  HL       ;HL=HL+1
LD   A,H      ;test iF
OR   A        ;HL=0
JR   Z,zero   ;iF so
               JR zero
```

IND

IND	237,170	ED AA	16

SZ H PNC
◆X ◆ ◆▮

◆ Changed
○ Res 0
▮ Set 1
X Special

```
IND is the equivelent
of doing the following
4 opcodes :-

IN   A,(C)
LD   (HL),A
DEC HL
DEC B

Z Flag SET to 1 iF B=0

Read port C,store at
address HL, DEC HL,
DEC B
```

INDR

INDR	237,186	ED BA	16	21

SZ H PNC
◆1 ◆ ◆▮

◆ Changed
○ Res 0
▮ Set 1
X Special

```
INDR is the equivelent
of doing the following
4 opcodes:-

Loop: IN   A,(C)
      LD   (HL),A
      DEC  HL
      DJNZ Loop

Read port C, store at
address HL, DEC HL, do
again number of B time
```

INI

INI	237,162	ED A2	16	SZ H PNC	◆ Changed
				◆X ◆ ◆!	◦ Res 0
					! Set 1
					X Special

INI is the equivelent of doing the following 4 opcodes :-

```
IN   A,(C)
LD  (HL),A
INC HL
DEC B
```

Z flag SET to 1 if B=0

Read port C, put at address HL, INC HL and DEC B

INIR

INIR	237,178	ED B2	16 21	SZ H PNC	◆ Changed
				◆! ◆ ◆!	◦ Res 0
					! Set 1
					X Special

INIR is the equivelent of doing the following 4 opcodes :-

```
Loop:  IN    A,(C)
       LD   (HL),A
       INC   HL
       DJNZ Loop
```

Read port C, store at address HL, INC HL, do again number of B time

JP s / JP flag,s (flag C,M,NC,PE,PO,Z)

JP nn	195,n,n	C3 n n	10	SZ H PNC	◆ Changed
C,nn	218,n,n	DA n n	10		◦ Res 0
M,nn	250,n,n	FA n n	10		! Set 1
NC,nn	210,n,n	D2 n n	10		X Special
NZ,nn	194,n,n	C2 n n	10		
P,nn	242,n,n	F2 n n	10		
PE,nn	234,n,n	EA n n	10		
PO,nn	226,n,n	E2 n n	10		
Z,nn	202,n,n	CA n n	10		
(HL)	233	E9	4		
(IX)	221,233,n,n	DD E9	8		
(IY)	253,233,n,n	FD E9	8		

JP is the equivelent to Sam's GOTO

```
JP 32768 ;Jump/Run/
         ;Goto 32768

JP (HL)  ;JP DPEEK HL
```

JR d / JR flag,d (flag C,NC,NZ,Z)

JR d	24,d	18 d	12	SZ H PNC	◆ Changed
C,d	56,d	38 d	12 7		◦ Res 0
NC,d	48,d	30 d	12 7		! Set 1
NZ,d	32,d	20 d	12 7		X Special
Z,d	40,d	28 d	12 7		

JR displacement from current opcode address JR backwards upto -126 or upto +129.

JR has the advantage over JP in that it is easy to relocate M/C to Run from a differnt address

LD A,r

A	127	7F	4
B	120	78	4
C	121	79	4
D	122	7A	4
E	123	7B	4
H	124	7C	4
L	125	7D	4
I	237,87	ED 57	9
R	237,95	ED 5F	9
n	62,n	3E n	7
IXh	221,124	DD 7C	8
IYh	253,124	FD 77	8
IXl	221,125	DD 7D	8
IYl	253,125	FD 7D	8

◆ Changed
◦ Res 0
! Set 1
X Special

SZ H PNC

```
A= reg

LD   A,B    ;A=B
LD   A,6    ;A=6
LD   A,IXL  ;A=IX reg
            ;Low part
```

LD A,(r)

(BC)	10	0A	7
(DE)	26	1A	7
(HL)	126	7E	7
(IX+d)	221,126,d	DD 7E d	19
(IY+d)	253,126,d	FD 7E d	19
(nn)	58,n,n	3A n n	13

◆ Changed
◦ Res 0
! Set 1
X Special

SZ H PNC

```
A= PEEK (reg)

LD A,(HL)   ;A= PEEK HL
            ;address
LD A,(IX+7) ;A= PEEK IX
            ;+7
LD A,(102)  ;A= PEEK
            ;address102
```

LD B,r

A	71	47	4
B	64	40	4
C	65	41	4
D	66	42	4
E	67	43	4
H	68	44	4
L	69	45	4
n	6,n	06 n	7
IXh	221,68	DD 44	8
IYh	253,68	FD 44	8
IXl	221,69	DD 45	8
IYl	253,69	FD 45	8
(HL)	70	46	7
(IX+d)	221,70,d	DD 46 d	19
(IY+d)	253,70,d	FD 46 d	19

◆ Changed
◦ Res 0
! Set 1
X Special

SZ H PNC

```
A= reg or PEEK (reg)

LD B,C      ;B=C
LD B,18     ;B=18
LD B,IYh    ;B=IY reg
            ;high part
LD B,(HL)   ;B= PEEK HL
LD B,(IY-9) ;B= PEEK IY
            ;-9
```

LD C,r

A	79	4F	4
B	72	48	4
C	73	49	4
D	74	4A	4
E	75	4B	4
H	76	4C	4
L	77	4D	4
n	14,n	0E n	7
IXh	221,76	DD 4C	8
IXh	253,76	FD 4C	8
IYl	221,77	DD 4D	8
IYl	221,77	FD 4D	8
(HL)	78	4E	7
(IX+d)	221,78,n	DD 4E d	19
(IX+d)	253,78,n	FD 4E d	19

◆ Changed
◦ Res 0
! Set 1
X Special

SZ H PNC

```
C= reg or PEEK (reg)

LD C,L      ;C=L
LD C,56     ;C=56
LD C,IXL    ;C= IX reg
            ;Low part
LD C,(HL)   ;C= PEEK HL
LD C,(IX+2) ;C= PEEK IX
            ;+2
```

LD D,d

A	87	57	4
B	80	50	4
C	81	51	4
D	82	52	4
E	83	53	4
H	84	54	4
L	85	55	4
n	22,n	16 n	7
IXh	221,84	DD 54	8
IYh	253,84	FD 54	8
IXl	221,85	DD 55	8
IYl	253,85	FD 55	8
(HL)	86	56	7
(IX+d)	221,86,d	DD 56 d	19
(IY+d)	253,86,d	FD 56 d	19

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
D= reg or PEEK (reg)

LD D,H        ;D= H
LD D,201      ;D= 201
LD D,IXL      ;D= IX reg
              ;Low part
LD D,(HL)     ;D= PEEK HL
LD D,(IY+0)   ;D= PEEK IY
              ;+0
```

LD E,r

A	95	5F	4
B	88	58	4
C	89	59	4
D	90	5A	4
E	91	5B	4
H	92	5C	4
L	93	5D	4
n	30,n	1E n	7
IXh	221,92	DD 5C	8
IYh	253,92	FD 5C	8
IXl	221,93	DD 5D	8
IYl	253,93	FD 5D	8
(HL)	94	5E	7
(IX+d)	221,94,d	DD 5E d	19
(IY+d)	253,94,d	FD 5E d	19

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
E= reg or PEEK (reg)

LD E,A        ;E= A
LD E,255      ;E= 255
LD E,IYh      ;E= IY reg
              ;high part
LD E,(HL)     ;E= PEEK HL
LD E,(IX-4)   ;E= PEEK IX
              ;-4
```

LD H,r

A	103	67	4
B	96	60	4
C	97	61	4
D	98	62	4
E	99	63	4
H	100	64	4
L	101	65	4
n	38,n	38,n	7
(HL)	102	66	7
(IX+d)	221,102,d	DD 66 d	19
(IY+d)	253,102,d	FD 66 d	19

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
H= reg or PEEK (reg)

LD H,L        ;H= L
LD H,0        ;H= 0
LD H,(HL)     ;H= PEEK HL
LD H,(IY-1)   ;H= PEEK IY
              ;-1
```

LD L,r

A	111	6F	4
B	104	68	4
C	105	69	4
D	106	6A	4
E	107	6B	4
H	108	6C	4
L	109	6D	4
n	46,n	2E n	7
(HL)	110	6E	7
(IX+d)	221,110,d	DD 6E d	19
(IY+d)	253,110,d	FD 6E d	19

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
L= reg or PEEK (reg)

LD L,E        ;L= E
LD L,13       ;L= 13
LD L,(HL)     ;L= PEEK HL
LD L,(IX-9)   ;L= PEEK IX
              ;-9
```

LD I or R,A

I,A	237,71	ED 47	9	SZ H PNC
R,A	237,79	ED 4F	9	

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD I,A ;I=A
LD R,A ;R=A
```

LD reg,nn

BC,nn	1,n,n	01 n n	10	SZ H PNC
DE,nn	17,n,n	11 n n	10	
HL,nn	33,n,n	21 n n	10	
SP,nn	49,n,n	31 n n	10	
IX,nn	221,33,n,n	DD 21 n n	14	
IY,nn	253,33,n,n	FD 21 n n	14	

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD reg,nn

LD HL,32768 ;HL=32768
```

LD reg,(nn)

BC,(nn	237,75,n,n	ED 4B n n	20	SZ H PNC
DE,(nn	237,91,n,n	ED 5B n n	20	
HL,(nn	42,n,n	2A n n	16	
SP,(nn	237,123,n,n	ED 7B n n	20	
IX,(nn	221,42,n,n	DD 2A n n	20	
IY,(nn	253,42,n,n	FD 2A n n	20	
A,(nn)	58 n n	3A n n	13	

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD reg,DPEEK addr

LD BC,(16384);BC=DPEEK
              ;16384

LD A,(49152)  ;A= PEEK
              ;49152
```

LD (reg),reg

(BC),A	2	02	7
(DE),A	18	12	7
(HL),A	119	77	7
(HL),B	112	70	7
(HL),C	113	71	7
(HL),D	114	72	7
(HL),E	115	73	7
(HL),H	116	74	7
(HL),L	117	75	7
(HL),n	54 n	36 n	10

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD (reg addr),reg

LD DE,49152 ;DE= 49152
LD (DE),A   ;POKE DE,A

LD HL,32768 ;HL= 32768
LD (HL),6   ;POKE HL,6
```

LD (nn),reg

(nn),A	50,n,n	32 n n	13
nn),BC	237,67,n,n	ED 43 n n	20
nn),DE	237,83,n,n	ED 53 n n	20
nn),HL	34,n,n	22 n n	16
nn),IX	221,34,n,n	DD 22 n n	20
nn),IY	253,34.n,n	FD 22 n n	20
nn),SP	237,115,n,n	ED 73 n n	20

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD (addr),reg

LD (40000),A ;POKE
             ;40000,A
LD (32768),HL;DPOKE
             ;32768,HL
```

LD IXh or l,reg

IXh,A	221,103	DD 67	8
IYh,A	253,103	FD 67	8
IXl,A	221,111	DD 6F	8
IYl,A	253,111	FD 6F	8
IXh,B	221,96	DD 60	8
IYh,B	253,96	FD 60	8
IXl,B	221,104	DD 68	8
IYl,B	253,104	FD 68	8
IXh,C	221,97	DD 61	8
IYh,C	253,97	FD 61	8
IXl,C	221,105	DD 69	8
IYl,C	253,105	FD 69	8

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD IX reg part,reg

LD IXL,B ;IX Low part
         ;=B
LD IYh,C ;IY high part
         ;=C
```

LD IXh or l,reg

IXh,D	221,98	DD 62	8
IYh,D	253,98	FD 62	8
IXl,D	221,106	DD 6A	8
IYl,D	253,106	FD 6A	8
IXh,E	221,99	DD 63	8
IYh,E	253,99	FD 63	8
IXl,E	221,107	DD 6B	8
IYl,E	253,107	FD 6B	8
IXh,n	221,38,n	DD 26 n	11
IYh,n	253,38,n	FD 26 n	11
IXl,n	221,46,n	DD 46 n	11
IYl,n	253,46,n	FD 46 n	11

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD IX reg part,reg or
            or number

LD IXh,C ;IX high part
         ;=C
LD IXL,8 ;IX Low part
         ;=8
```

LD IXh or l,IXh or l

IXhIXh	221,100	DD 64	8
IYhIYh	253,100	FD 64	8
IXhIXl	221,101	DD 65	8
IYhIXl	253,101	FD 65	8
IXlIXh	221,108	DD 6C	8
IYlIYh	253,108	FD 6C	8
IXlIXl	221,109	DD 6D	8
IYlIYl	253,109	FD 6D	8

SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
LD IXh/L,IXh/L

LD IYh,IXL;IY high
          ;part =
          ;IX Low part
```

LDD

LDD	237,168	ED A8	16	SZ H PNC	◆ Changed ○ Res 0 ▮ Set 1 X Special
				○ X○	

LDD is the equivelent of doing the following 5 opcodes: -

```
LD   A,(HL)  ;transfer
LD  (DE),A   ;POKE DE,
DEC HL       ;PEEK HL
DEC DE
DEC BC
```

If BC=0 then P flag=0

LDDR

LDDR	237,184	ED B8	16 21	SZ H PNC	◆ Changed ○ Res 0 ▮ Set 1 X Special
				○ ○○	

LDDR is the equivelent of doing the following 8 opcodes :-

```
Loop: LD   A,(HL)
      LD  (DE),A
      DEC HL
      DEC HL
      DEC BC
      LD   A,B
      OR   C
      JR   NZ,Loop
```

transfer large areas of memory

LDI

LDI	237,160	ED A0	16	SZ H PNC	◆ Changed ○ Res 0 ▮ Set 1 X Special
				○ X○	

LDI is the equivelent of doing the following 5 opcodes :-

```
LD   A,(HL)  ;transfer
LD  (DE),A   ;POKE DE,HL
INC HL
INC DE
DEC BC
```

If BC=0 then P flag =0

LDIR

LDIR	237,176	ED B0	16 21	SZ H PNC	◆ Changed ○ Res 0 ▮ Set 1 X Special
				○ ○○	

LDIR is the equivelent of doing the following 8 opcodes :-

```
Loop: LD   A,(HL)
      LD  (DE),A
      INC HL
      INC DE
      DEC BC
      LD A,B
      OR C
      JR NZ,Loop
```

Transferes large areas of memory

NEG

NEG	237,68	ED 44	8	SZ H PNC
				◆◆ ◆ ◆!◆

◆ Changed
◦ Res 0
! Set 1
X Special

```
A= toggle - or +

LD A,16 ;A=+16
NEG     ;A=-16

LD A,-1 ;A=-1
NEG     ;A=+1

C flag=0 if A reg was
  a 0 before the NEG
  was Run
C flag=0 and P flag=1
if A reg was 128
before NEG was Run
```

NOP

NOP	0	00	4	SZ H PNC

◆ Changed
◦ Res 0
! Set 1
X Special

```
NOP does nothing, just
acts as a Pause for 4
Tstates. Can be used
to cancel Machine code
instructions form say
an Assembler Source
File e.g

LD  HL,32768
DEC (HL) ;DEC (HL) is
RET      ;a one byte
opcode so replace this
with NOP or DB 0 to
cancell DEC (HL)
```

OR s

(HL)	182	B6	7	SZ H PNC
A	183	B7	4	
B	176	B0	4	◆◆ ◦ ◆◦◦
C	177	B1	4	
D	178	B2	4	
E	179	B3	4	
H	180	B4	4	
L	181	B5	4	
n	246,n	F6 n	7	
IXh	221,180	DD B4	8	
IYh	253,180	FD B4	8	
IXl	221,181	DD B5	8	
IYl	253,181	FD B5	8	
(IX+d)	221,182,d	DD B6 d	19	
(IY+d)	253,221,d	FD B6 d	19	

◆ Changed
◦ Res 0
! Set 1
X Special

```
A= A or reg

LD A,%00101101
LD H,%01110001
OR H
A now=01111101
Each bit is compared
by both regs and PUT
in the A reg, bit RES
to 0 only if there are
two 0's others SET to1
OR %11110000 will SET
bits 7,6,5,4 and leave
bits 3,2,1,0 alone.
```

OTDR

OTDR	237,187	ED BB	16	21	SZ H PNC
					◆! ◆ ◆!

◆ Changed
◦ Res 0
! Set 1
X Special

```
OTDR is the equivelent
of doing the following
4 opcodes :-

Loop: LD  A,(HL)
      OUT (C),A
      DEC HL
      DJNZ Loop

LD HL,colookuptable
LD B,16   ;Altering
LD C,248  ;the colour
OTDR      ;Look up
table on Sam
```

OTIR

OTIR	237,179	ED B3	16	21	SZ H PNC
					♦! ♦ ♦!

◆ Changed
◦ Res 0
! Set 1
X Special

```
OTIR is the equivelent
of doing the following
4 opcodes :-

Loop: LD A,(HL)
      OUT (C),A
      INC HL
      DJNZ Loop
```

OUT (C),reg or n

(C),A	237,121	ED 79	12	SZ H PNC
(C),B	237,65	ED 41	12	
(C),C	237,73	ED 49	12	
(C),D	237,81	ED 51	12	
(C),E	237,89	ED 59	12	
(C),H	237,97	ED 61	12	
(C),L	237,105	ED 69	12	
(n),A	211,n	D3 n	11	

◆ Changed
◦ Res 0
! Set 1
X Special

```
Ouput  reg to port C

IN  A,(251)  ;A= PEEK
             ;port 251
INC A        ;INC A
OUT(251),A   ;Output
             ;A reg in
             ;port A
LD  C,251    ;Output A
OUT(C),A     ;reg in
             ;port C
```

OUTD

OUTD	237,71	ED AB	16	SZ H PNC
				♦X ♦ ♦!

◆ Changed
◦ Res 0
! Set 1
X Special

```
OUTD is the equivelent
of doing the following
4 opcodes :-

LD  A,(HL)
OUT (C),A
DEC HL
DEC B

Z Flag SET to 1,if B=0
```

OUTI

OUTI	237,163	ED A3	16	SZ H PNC
				♦X ♦ ♦!

◆ Changed
◦ Res 0
! Set 1
X Special

```
OUTI is the equivelent
of doing the following
4 opcode :-

LD  A,(HL)
OUT(C),A
INC HL
DEC B

Z Flag SET to 1,if B=0
```

POP reg

AF	241	F1	10
BC	193	C1	10
DE	209	D1	10
HL	225	E1	10
IX	221,225	DD E1	14
IY	253,225	FD E1	14

SZ H PNC

◆ Changed
◦ Res 0
ı Set 1
X Special

GET 16 bit number of the stack and store it in register.

```
POP HL ;HL= value of
       ;the stack
       ;pointed by SP
```

PUSH reg

AF	245	FD E1	11
BC	197	F5	11
DE	213	C5	11
HL	229	D5	11
IX	221,229	E	11
IY	253,229	DD E5	15
		FD E5	15

SZ H PNC

◆ Changed
◦ Res 0
ı Set 1
X Special

PUT 16 bit register on stack

```
LD HL,40000 ;HL=40000
PUSH HL     ;40000 on
            ;stack
```

PUSH can be useful to store numbers temp,use POP to get value back Be careful to keep the PUSHes and POPs evenly balanced on stack else reset on ret to Basic

RES b,s (b is a number from 0-7)

b,(HL)	203,134+(b*8)	CB 86+(b*08	15
b,A	203,135+(b*8)	CB 87+(b*08	8
b,B	203,128+(b*8)	CB 80+(b*08	8
b,C	203,129+(b*8)	CB 81+(b*08	8
b,D	203,130+(b*8)	CB 82+(b*08	8
b,E	203,131+(b*8)	CB 83+(b*08	8
b,H	203,132+(b*8)	CB 84+(b*08	8
b,L	203,133+(b*8)	CB 85+(b*08	8
b,(IXd	221,203,d,134+(b*8)	DD CB d 86+(b*08	23
b,(IYd	253,203,d,134+(b*8)	FD CB d 86+(b*08	23

SZ H PNC

◆ Changed
◦ Res 0
ı Set 1
X Special

RES bit of register

```
RES 1,B ;RES bit 1 of
        ;B reg
RES 7,(IX+3) ;RES bit7
             ;of IX
             ;reg +3
```

RET

RET	201	C9	10
RET C	216	D8	11 5
RET M	248	F8	11 5
RET NC	208	D0	11 5
RET NZ	192	C0	11 5
RET P	240	F0	11 5
RET PE	232	E8	11 5
RET PO	224	E0	11 5
RET Z	200	C8	11 5
RETI	237,77	ED 4D	14
RETN	237,69	ED 45	14

SZ H PNC

◆ Changed
◦ Res 0
ı Set 1
X Special

RET is like a RETURN Sam Basic command.

RET POPs the address on the stack and then Jumps-goto that addr.

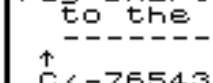
```
DEC A ;RETs only if A
RET Z ;reg= 0 only
EI    ;Return form a
RETI  ;Sam mouse inte-
      ;rrupt routine
RETN  ;Rteurn from a
a non mask interrupt
```

RL s

(HL)	203,22	CB 16	15	SZ H PNC
A	203,23	CB 17	8	◆◆ ○ ◆○◆
B	203,16	CB 10	8	
C	203,17	CB 11	8	
D	203,18	CB 12	8	
E	203,19	CB 13	8	
H	203,20	CB 14	8	
L	203,21	CB 15	8	
(IX+d)	221,203,d,22	DD CB d 16	23	
(IY+d)	253,203,d,22	FD CB d 16	23	

◆ Changed
○ Res 0
! Set 1
X Special

reg=shift bits in reg
to the Left



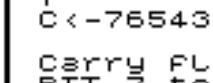
Carry flag to Bit 0
BIT 7 to Carry
BIT 6 to BIT 7
BIT 5 to BIT 6
etc etc
BIT 0 to BIT 1

RLA

RLA	23	17	4	SZ H PNC
				○ ○◆

◆ Changed
○ Res 0
! Set 1
X Special

A reg bits are shifted
to the Left



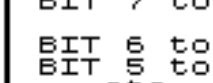
Carry flag to BIT 0
BIT 7 to Carry
BIT 6 to BIT 7
BIT 5 to BIT 6
etc etc
BIT 0 to BIT 1

RLCA

RLCA	7	07	4	SZ H PNC
				○ ○◆

◆ Changed
○ Res 0
! Set 1
X Special

A reg bits moved to
the Left



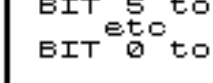
BIT 7 to Carry and
BIT 0
BIT 6 to BIT 7
BIT 5 to BIT 6
etc etc
BIT 0 to BIT 1

RLC r

(HL)	203,6	CB 06	15	SZ H PNC
A	203,7	CB 07	8	◆◆ ○ ◆○◆
B	203,0	CB 00	8	
C	203,1	CB 01	8	
D	203,2	CB 02	8	
E	203,3	CB 03	8	
H	203,4	CB 04	8	
L	204,5	CB 05	8	
(IX+d)	221,203,d,6	DD CB d 06	23	
(IY+d)	253,203,d,6	FD CB d 06	23	

◆ Changed
○ Res 0
! Set 1
X Special

reg move bits to the
Left



BIT 7 to Carry flag
and BIT 0
BIT 6 to BIT 7
BIT 5 to BIT 6
etc etc
BIT 0 to BIT 1

RLD

RLD	203,111	ED 6F	18	SZ H PNC

◆ Changed
◦ Res 0
ı Set 1
× Special

◆◆ ◦ ◆◦

```
Shift A Reg & PEEK HL
          -----------
     A  ↑       (HL) |
A7654 3210 <-7654 3210
                |    |
                -----
A reg flags to PEEK
                  (HL+1)
PEEK (HL+1) to PEEK
                  (HL+0)
PEEK (HL+0) to A reg
                  flags
```

RR s

(HL)	203,30	CB 1E	15	SZ H PNC
A	203,31	CB 1F	8	
B	203,24	CB 18	8	
C	203,25	CB 19	8	
D	203,26	CB 1A	8	
E	203,27	CB 1B	8	
H	203,28	CB 1C	8	
L	203,29	CB 1D	8	
(IX+d)	221,203,d,30	DD CB d 1E	23	
(IY+d)	253,203,d,30	FD CB d 1E	23	

◆ Changed
◦ Res 0
ı Set 1
× Special

◆◆ ◦ ◆◦◆

```
Shift bits in register
to the right
     ---------
    |         ↑
    76543210->C

 Carry flag to BIT 7
 BIT 0 to Carry flag
 BIT 1 to BIT 0
 BIT 2 to BIT 1
    etc  etc
 BIT 7 to BIT 6
```

RRA

RRA	31	1F	4	SZ H PNC

◆ Changed
◦ Res 0
ı Set 1
× Special

◦ ◦◆

```
Shift bits in A reg to
  --------   the right
 |         ↑
 76543210->C

 Carry flag to BIT 7
 BIT 0 to Carry flag
 BIT 1 to BIT 0
 BIT 2 to BIT 1
    etc  etc
 BIT 7 to BIT 6
```

RRC s

(HL)	203,14	CB 0E	15	SZ H PNC
A	203,15	CB 0F	8	
B	203,8	CB 08	8	
C	203,9	CB 09	8	
D	203,10	CB 0A	8	
E	203,11	CB 0B	8	
H	203,12	CB 0C	8	
L	203,13	CB 0D	8	
(IX+d)	221,203,d,14	DD CB d 0E	23	
(IY+d)	253,203,d,14	FD CB d 0E	23	

◆ Changed
◦ Res 0
ı Set 1
× Special

◆◆ ◦ ◆◦◆

```
Shift bits in reg to
              the right
  ---------
 |         ↑
 76543210->C

 BIT 0 to Carry flag
 BIT 0 to BIT 7
 BIT 1 to BIT 0
 BIT 2 to BIT 1
    etc  etc
 BIT 7 to BIT 6
```

RRCA

RRCA	15	0F	4	SZ H PNC ◦ ◦◆

◆ Changed
◦ Res 0
! Set 1
X Special

```
Shift Bits in A reg to
                the right
  ---------
 |         ↑
 76543210->C

 BIT 0 to Carry Flag
 BIT 0 to BIT 7
 BIT 1 to BIT 0
 BIT 2 to BIT 1
    etc  etc
 BIT 7 to BIT 6
```

RRD

RRD	237,103	ED 67	18	SZ H PNC ◆◆ ◦ ◆◦

◆ Changed
◦ Res 0
! Set 1
X Special

```
Shift A reg Flags and
                PEEK HL
          ----  ----
         |    | ↑    |

 7654 3210 7654 3210
      ----      ----
       |          |
        ---------
 PEEK (HL+1) to A reg
                  Flags
 A reg Flags to PEEK
                   (HL)
 PEEK (HL) to (HL+1)
```

RST n

0	199	C7	11	SZ H PNC
8	207	CF	11	
16	215	D7	11	
24	223	DF	11	
32	231	E7	11	
40	239	EF	11	
48	247	F7	11	
56	255	FF	11	

◆ Changed
◦ Res 0
! Set 1
X Special

```
RST is a 1 byte opcode
which will Jump to one
of 8 places in the Rom

RST 0 ;JP 0
RST 8 ;JP 8
RST 16;JP 16
RST 24;JP 24
RST 32;JP 32
RST 40;JP 40
RST 48;JP 48
RST 56;JP 56
```

SBC A,s

(HL)	158	9E	7	SZ H PNC
A	159	9F	4	
B	152	98	4	
C	153	99	4	
D	154	9A	4	
E	155	9B	4	
H	156	9C	4	
L	157	9D	4	
n	222,n	DE n	4	
IXh	221,156	DD 9C	8	
IYh	253,156	FD 9C	8	
IXl	221,157	DD 9D	8	
IYl	253,157	FD 9D	8	
(IX+d)	221,158,d	DD 9E d	19	
(IY+d)	253,158,d	FD 9E d	19	

◆ Changed
◦ Res 0
! Set 1
X Special

```
A= (reg+carry)-A

LD   H,10 ;H=10
LD   A,9  ;A=9
SBC  A,H  ;A=H+carry-A
          ;A now = 1
```

SBC HL,nn

BC	237,66	ED 42	15	SZ H PNC
DE	237,82	ED 52	15	◆◆ ◆ ◆↕◆
HL	237,98	ED 62	15	
SP	237,114	ED 72	15	

◆ Changed
◦ Res 0
↕ Set 1
X Special

```
HL= Reg+carry-HL

LD   HL,49152 ;HL=49152
LD   DE,32768 ;DE=32768
AND  A        ;RES 0
              ;carry
SBC  HL,DE    ;HL=HL+
              ;carry-DE
JR   NC,under ;IF HL=DE
              ;or HL<DE
              ;then JR
              ;under
```

SCF

SCF	55	37	4	SZ H PNC
				◦ ◦↕

◆ Changed
◦ Res 0
↕ Set 1
X Special

```
Carry flag is SET to 1
```

SET flag,s (flag is 7,6,5,4,3,2,1,0)

b,(HL)	203,198+(b*8)	CB C6+(b*08	15	SZ H PNC
b,A	203,199+(b*8)	CB C7+(b*08	8	
b,B	203,192+(b*8)	CB C0+(b*08	8	
b,C	203,193+(b*8)	CB C1+(b*08	8	
b,D	203,194+(b*8)	CB C2+(b*08	8	
b,E	203,195+(b*8)	CB C3+(b*08	8	
b,H	203,196+(b*8)	CB C4+(b*08	8	
b,L	203,197+(b*8)	CB C5+(b*08	8	
b,(IXd	221,203,d,198+(b*8)	DD CB d C6+(b*08)	23	
b,(IYd	253,203.d.198+(b*8)	FD CB d C6+(b*08)	23	

◆ Changed
◦ Res 0
↕ Set 1
X Special

```
SET bit of reg to 1

SET 3,A ;SET BIT 3 of
        ;A reg to 1

SET 7,(HL) ;SET BIT 7
           ;of PEEK
           ;(HL) to 1
```

SLA s

(HL)	203,38	CB 26	15	SZ H PNC
A	203,39	CB 27	8	◆◆ ◦ ◆◦◆
B	203,32	CB 20	8	
C	203,33	CB 21	8	
D	203,34	CB 22	8	
E	203,35	CB 23	8	
H	203,36	CB 24	8	
L	203,37	CB 25	8	
(IX+d)	221,203,d,38	DD CB d 26	23	
(IY+d)	253,203,d,38	FD CB d 26	23	

◆ Changed
◦ Res 0
↕ Set 1
X Special

```
Shift bits in reg to
the Left, RES BIT 0,0


 C<-76543210<-0

 BIT 7 to Carry Flag
 BIT 6 to BIT 7
 BIT 5 to BIT 6
    etc  etc
 BIT 1 to BIT 0
 RES 0 BIT 0
```

SRA s

(HL)	203,46	CB 2E		15
A	203,47	CB 2F		8
B	203,40	CB 28		8
C	203,41	CB 29		8
D	203,42	CB 2A		8
E	203,43	CB 2B		8
H	203,44	CB 2C		8
L	203,45	CB 2D		8
(IX+d)	221,203,d,46	DD CB d 2E		23
(IY+d)	253,203,d,46	FD CB d 2E		23

SZ H PNC
♦♦ ○ ♦○♦

♦ Changed
○ Res 0
! Set 1
X Special

```
Reg =shift bits in reg
   to the right


|>76543210->C
|-|

 BIT 0 to Carry Flag
 BIT 1 to BIT 0
 BIT 2 to BIT 1
    etc  etc
 BIT 6 to BIT 5
 BIT 7 Leave alone
```

SRL s

(HL)	203,62	CB 3E		15
A	203,63	CB 3F		8
B	203,56	CB 38		8
C	203,57	CB 39		8
D	203,58	CB 3A		8
E	203,59	CB 3B		8
H	203,60	CB 3C		8
L	203,70	CB 3D		8
(IX+d)	221,203,d,62	DD CB d 3E		23
(IY+d)	253,203,d,62	FD CB d 3E		23

SZ H PNC
♦♦ ○ ♦○♦

♦ Changed
○ Res 0
! Set 1
X Special

```
reg= shift bits in reg
     to the right


 0->76543210->C

 RES 0     BIT 7
 BIT 6 to BIT 5
 BIT 5 to BIT 4
    etc  etc
 BIT 1 to BIT 0
 BIT 0 to Carry Flag
```

SUB s

(HL)	150	96		7
A	151	97		4
B	144	90		4
C	145	91		4
D	146	92		4
E	147	93		4
H	148	94		4
L	149	95		4
n	214,n	D6 n		7
IXh	221,148	DD 9C		8
IYh	253,148	FD 9C		8
IXl	221,149	DD 9D		8
IYl	253,149	FD 9D		8
(IX+d)	221,150,d	DD 96 d		19
(IY+d)	253,150,d	FD 96 d		19

SZ H PNC
♦♦ ♦ ♦!♦

♦ Changed
○ Res 0
! Set 1
X Special

```
A= reg-A

LD   H,90  ;H=90
LD   A,56  ;A=56
SUB H      ;A=A-H
           ;now =34
```

XOR s

(HL)	174	AE		7
A	175	AF		4
B	168	A8		4
C	169	A9		4
D	170	AA		4
E	171	AB		4
H	172	AC		4
L	173	AD		4
n	238,n	EE n		7
IXh	221,172	DD AC		8
IYh	253,172	FD AC		8
IXl	221,173	DD AD		8
IYl	253,173	FD AD		8
(IX+d)	221,74,d	DD AE d		19
(IY+d)	253,74,d	FD AE d		19

SZ H PNC
♦♦ ○ ♦○○

♦ Changed
○ Res 0
! Set 1
X Special

```
A= A XOR reg

LD   A,%01010011
LD   H,%10011010
XOR H= 11011011

BITS in reg and A reg
are XORed (Merged)
IF both bits are 0 the
n RES BIT to 0, others
 SET to 1
XOR A is a short and
quick way to LD A,0
```