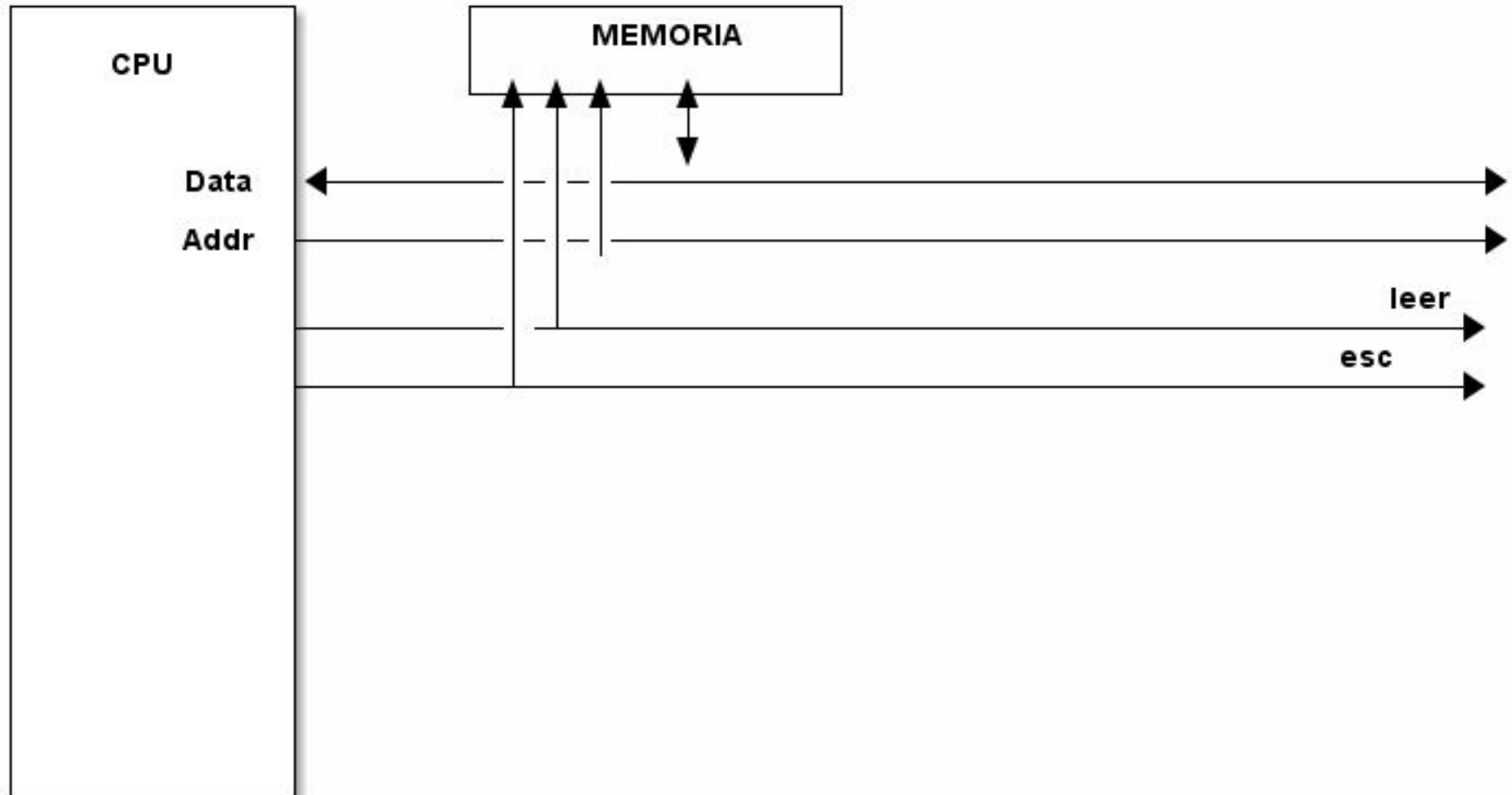


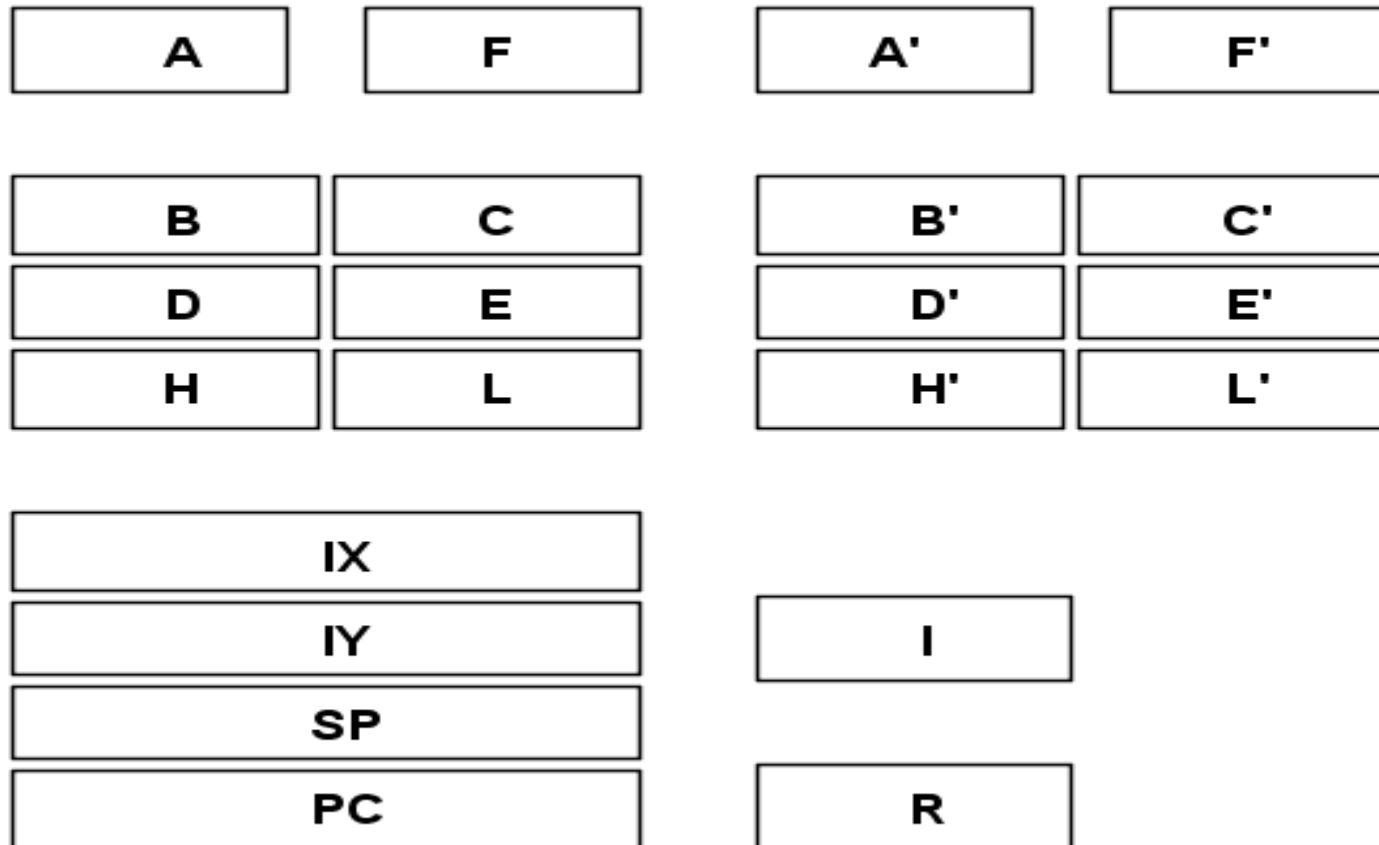
Repaso - Z80



Repaso – Formato instrucciones

- 4 direcciones
 - Opcode oper1 oper2 result prox_instrucc
- Instrucciones de operación o de bifurcación
 - Registro PC para recordar direcc. de prox. Instr.
- Acumulador
 - $A \leftarrow A \text{ op } \text{oper2}$

Repaso – Registros Z80



Modos de direccionamiento

- Las distintas maneras de especificar en una instrucción de **dónde obtener** el valor de los **operandos**, el lugar donde almacenar el **resultado** o dónde ir a buscar la **siguiente instrucción** a ejecutar.
 - Implícito
 - Inmediato
 - Registro
 - Directo
 - Indirecto por Registro
 - Indexado
 - De a bit
 - Relativo al programa
 - Página 0

Modos de direccionamiento

- Implícito

- Implícito en el propio código de operación.
- Se ejecuta siempre sobre el mismo operando.
- CPL ; complementa bits del acum,

0xffff	
	...
	OPCODE
	...
0x0000	

Modos de direccionamiento

- Inmediato (8 y 16 bits)
 - Operando es una cte.
 - Viene a continuación de opcode.
 - ADD A, 27

0xffff	
	...
	Opcode
	27
	OPCODE
	...
0x0000	

Modos de direccionamiento

- Registro

- Operando es el contenido de un registro interno.
- Algunos bits del opcode determinan cuál es el registro.
- AND reg
- p. ej. AND reg. El opcode es 1000 0rrr dónde rrr identifica un reg (A, B, C, D, E, H, L)

0xffff	
	...
	...
	Opcode
	OPCODE
	...
0x0000	

Modos de direccionamiento

- Directo

- La dirección del operando se suministra a continuación del opcode.
- LD A, (dir)
- dir es una dirección de 16 bits

0xffff	
	...
	...
	dirH
	dirL
	OPCODE
	...
0x0000	

Modos de direccionamiento

- Indirecto por registro
 - La dirección del operando es el contenido de un par de registros.
 - Permite **calcular** direcciones.
 - LD A, (dir)
 - dir es una dirección de 16 bits

0xffff	
	...
	...
	...
	opcode
	OPCODE
	...
0x0000	

Modos de direccionamiento

- Indexado
 - La dirección se forma sumando el contenido de uno de los registros índice (IX o IY) con un desplazamiento de 8 bits.
 - Despl en complemento a 2, por tanto puede valer entre -127 y 128..
 - INC (IX+5)

Repertorio de instrucciones

- 158 instrucciones
- 1, 2 y en algunos casos 3 bytes de OP-CODE
- Datos adicionales:
 - dato inmediato, dirección de 2 bytes, desplazamiento en direccionamiento relativo.
- Compatibilidad con 8080 de Intel

Repertorio de instrucciones

- Cartilla:
 - Mnemonic
 - Operación: descripción similar a RTL.
 - Flags: indicación de cómo afecta a cada uno de los flags.
 - OPCODE: en binario y en hexadecimal.
 - Bytes: cantidad de bytes que ocupa en memoria.
 - Ciclos M: cantidad de ciclos de máquina necesarios para ejecutarla.
 - T States: cantidad de períodos de reloj que dura la ejecución.

Transferencias

- Varios modos de direccionamiento
- Nunca de memoria a memoria
- Variaciones de instrucción Load (LD)

Transferencias

- Mover 1 byte desde la dirección origen a la dirección destino.

```
LD A, (origen)
LD (destino), A
```

- Mover 2 bytes

```
LD HL, origen ; inmediato
LD DE, destino
LD A, (HL)
LD (DE), A

INC HL
INC DE
LD A, (HL) ; Indirecto
                ; por registro
LD (DE), A
```

Transferencias

- Stack
 - “crece” hacia abajo.
 - SP apunta al último ocupado
 - Transferencias 16 bits
- Instrucciones
 - Push reg16
 - Pop reg16
- Utilización para preservar contexto

```
push AF
push BC
...
pop AF
pop BC
```

Transferencias

- Ejemplo. Al inicio:

- BC = 00 01
- DE = 0B 07
- HL = 03 FF

```
...  
PUSH BC  
PUSH HL  
POP DE  
...
```


Lógicas y aritméticas de 8 bits

- Acumulador y ALU
- Banderas
 - Acarreo y préstamo
 - Overflow
- Comparaciones (cp)
- Notación cartilla
 - En la suma (add), un renglón para cada modo de direccionamiento
 - Para el resto un solo renglón
- ADD / ADC
- SUB / SBC
- AND / OR / XOR
- CP
- INC / DEC

Lógicas y aritméticas de 8 bits

- Ejemplo:
 - suma de HL con BC ($HL \leftarrow HL + BC$)
 - Suma los menos significativos y afectando flag Carry
 - Suma con acarreo los más significativos

```
ld a, l
add c      ; a ← a + c
ld l, a
ld a, h
adc b      ; a ← a + b + Cy
ld h, a
```

Aritméticas de propósito general y control de CPU

- Que operan sobre A
 - NEG / CPL
- Que operan sobre flag Cy
 - SCF / CCF
- Relleno, detención
 - NOP / HALT
- Ajuste de suma decimal
 - DAA
- Control de interrupciones
 - DI / EI / IM 0 / IM 1 / IM 2

Aritméticas de 16 bits

- Sumas, restas, incrementos, decrementos
- INC y DEC no afectan flags
 - Trucos, p. ej. para saber si HL es cero
 - `ld a, l / or h`
- No hay instrucción que copie SP a otro registro
 - Trucos:
 - O bien: `ld (dir), sp / ld hl, (dir)`
 - O bien: `ld hl, 0 / add hl, sp`

Rotación y desplazamiento

- RLC (Rotate Left Circular)
 - 8 bits, Cy fuera del anillo
- RL (Rotate Left)
 - 9 bits, Cy incluido.
- SRA (Shift Right Arithmetic)
 - División entera por 2
- SRL (Shift Right Logic)
- SLA (Shift Left Arithmetic)
- Ejemplo: multiplicar por 10
 - $A \leftarrow B * 10$
 - $A \leftarrow (B * 8) + (B * 2)$

```
SLA B ; x 2
```

```
LD A, B
```

```
SLA B ; x 4
```

```
SLA B ; x 8
```

```
ADD B
```

Salto, Llamadas y Retornos

- Modifican el PC
- Salto (Jump)
 - Incondicional: JP dir
 - Condicional: JP Z, dir
 - En el ejemplo salta si fue 0 el resultado de la última operación (la bandera Z está “preendida”)
- Salto
 - Absoluto (JP): $PC \leftarrow dir$
 - ocupa 3 bytes (opcode, dirL, dirH)
 - Relativo (JR): $PC \leftarrow PC + desp8$
 - ocupa solo 2 bytes
 - Reubicable
 - El ensamblador se encarga

Saltos, Llamadas y Retornos

Ejemplo

- Mover N bytes

- B: N bytes a mover
- HL: origen
- DE: destino

```
LD B, cant
LD HL origen
LD DE, destino
repetir:
LD A, (HL)
LD (DE), A
INC HL
INC DE
DEC B
JR NZ, repetir
```

Saltos, Llamadas y Retornos

- CALL dir
 - Permite llamar al mismo trozo de código desde diferentes partes de un programa.
- RET
 - Retorna a la instrucción siguiente a CALL
- Mecanismo
 - CALL guarda dirección de retorno en el stack.
 - Sería equivalente a
 - “PUSH PC”
 - JP dir
 - RET toma dato del stack y lo carga en contador de programa
 - Sería equivalente a:
 - “POP PC”
 -

Saltos, Llamadas y Retornos

Ejemplo

- Mismo ejemplo como subrutina

- Prog. ppal.:

```
...  
LD B, cant1  
LD HL origen1  
LD DE, destino1  
CALL mover  
....  
LD B, cant2  
LD HL origen2  
LD DE, destino2  
CALL mover
```

mover:

```
LD A, (HL)  
LD (DE), A  
INC HL  
INC DE  
DJNZ mover  
ret
```

Entrada / Salida

- Hardware
 - Se activa señal de control /IORQ
 - Valen solamente los 8 bits menos significativos de direcciones (A7..A0)
 - Solo 256 puertos de entrada y 256 de salida
- Instrucciones
 - Direccionamiento directo (solo al acumulador)
 - IN A, (dir8) / OUT (dir8), A
 - Direccionamiento indirecto con registro C
 - IN reg, (C) / OUT (C), reg
 - Transferencias en bloque