

LECTURAS SOBRE COMPUTADORAS DIGITALES –LECTURA N°9 **MATERIA: ARQUITECTURA DE LAS COMPUTADORAS**

1.- DEFINICIÓN DE MICROPROCESADOR

Todas las computadoras actuales se basan en un circuito integrado de silicio (chip de silicio) denominado microprocesador (uP) que posee en su interior la Unidad de Control, el ALU y algunos Registros de uso general de una computadora digital.

Los uP se diseñaron a partir de definir el formato de sus instrucciones, y los criterios utilizados fueron distintos a los expuestos en la Lectura N° 5, en la misma, todas las instrucciones tenían el mismo formato y, además, direccionaban en forma directa a los datos.

Recordemos que el Contador de Programa direcciona instrucciones y que las instrucciones direccionan datos. El Contador de Programa siempre direcciona en forma directa a las instrucciones, por este motivo su tamaño o el Bus de Direcciones asociado, es el que limita el tamaño máximo de memoria.

La base de diseño fue la arquitectura de la IBM /360 que introduce el direccionamiento indirecto y el direccionamiento inmediato, entre otros conceptos.

Primeramente veamos un poco de historia del uP.

1.1.- Un poco de historia

Los uP fueron creados en los Laboratorios Intel de Estados Unidos por pedido de una empresa japonesa, Busicom, que deseaba fabricar calculadoras de mano.

A cargo del diseño de base estuvo *Marcian Edward Ted Hoff, Jr.*, desde 1969 pero viendo que no se cumplían los plazos del contrato, se convocó al científico italiano Federico Faggin, que en 1971, más precisamente el 30 de noviembre, permitió lanzar al mercado el primer uP.

El mismo se denominó Intel 4004, por el tamaño de la palabra lógica de 4 bits, sus especificaciones técnicas son:

- Microprocesador de 4 bits
- Contiene 2.300 transistores
- Encapsulado de 16 pines
- Máxima velocidad del reloj 740 KHz
- Usa Arquitectura Harvard, es decir, almacenamiento separado de programas y datos. Contrario a la mayoría de los diseños con arquitectura de Harvard, que utilizan buses separados, el 4004, con su necesidad de mantener baja la cuenta de pines, usaba un bus de 4 bits multiplexado para transferir:
- 12 bits de direcciones (direccionando hasta 4 KB)
- Instrucciones de 8 bits de ancho, que no deben ser colocadas en la misma memoria de datos de 4 bits de ancho.
- El conjunto de instrucciones está formado por 46 instrucciones (de las cuales 41 son de 8 bits de ancho y 5 de 16 bits de ancho)

- 16 registros de 4 bits cada uno

Antes de continuar con el análisis de las características técnicas del 4004, veamos el método de razonamiento de Faggin.

Tomando el esquema de una computadora, ver figura 1, convirtió la Unidad de Entrada en un Teclado Numérico/Funcional(que incluía suma, resta, multiplicación, división), (ALU+Control) se transforman en el uP, la memoria Principal se convierte en dos memorias, una de programas fijos, memoria ROM y otra de datos, la memoria RAM, la unidad de salida será un display y la memoria asociada para mantener la información, y se obtiene el esquema de la Figura 2.

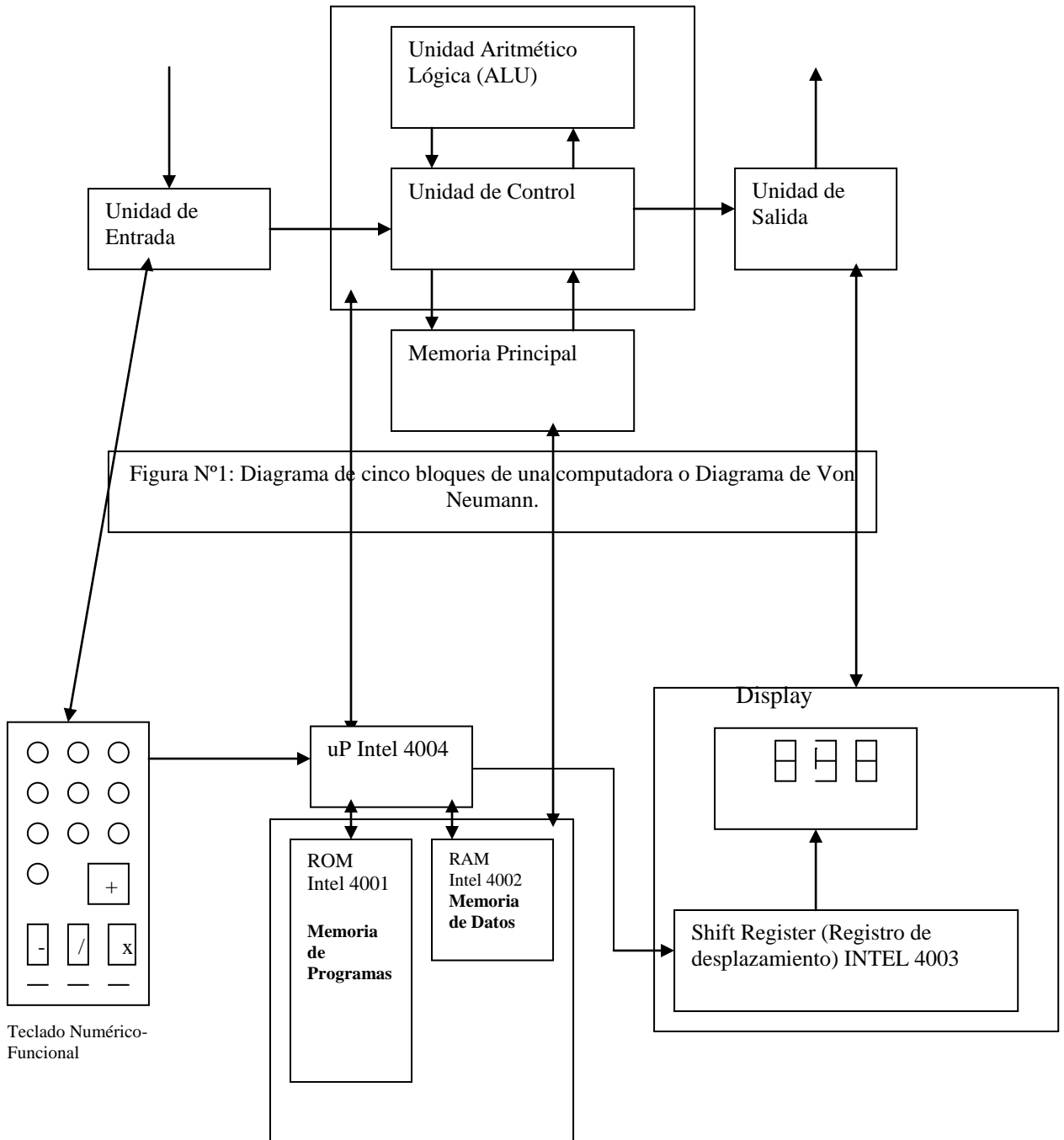


Figura 2: Esquema de la calculadora creada por Intel y el concepto de “familia” de circuitos integrados

El microprocesador Intel 4004 forma parte de una familia que le permite cumplir la función de calculadora, a esos miembros de la "familia" se los denomina "chips de soporte" o chipset, los mismos son:

- 4001: ROM de 256 bytes (256 instrucciones de programa de 8bits), y un puerto incorporado de I/O de 4 bits*
- 4002: RAM de 40 bytes (80 palabras de datos de 4 bits), y un puerto de salida incorporado de 4 bits. La porción de RAM del chip está organizada en cuatro "registros" de veinte palabras de 4 bits:
- 16 palabras datos (usadas para los dígitos de mantisa en el diseño original de la calculadora)
- 4 palabras de estado (usadas para los dígitos de exponente en el diseño original de la calculadora)
- 4003: shift register (registro de desplazamiento) de salida paralela de 10 bits para explorar teclados, pantallas, impresoras, etc.

Las características del 4004, ya listadas, son:

- Microprocesador de 4 bits: *Indica el tamaño de la palabra lógica, a partir de este primer microprocesador, todos se clasificaron, en principio, por esta característica, de esta manera se tienen uP de 8, 16, 32 y 64 bits.*
- Contiene 2.300 transistores: *Los actuales micro integran millones de transistores, se ha recorrido un largo camino, sin duda.*
- Encapsulado de 16 pines: *El número de pines ha crecido hasta 256 o más...*
- Máxima velocidad del reloj 740 KHz: *Con esta frecuencia de reloj, se puede calcular el tiempo de duración de un pulso:*

$$T = 1/f \quad \longrightarrow \quad T = \frac{1}{740 \times 10^3 \text{ pulsos/seg}} \quad \longrightarrow \quad T = \frac{1000 \times 10^{-6} \text{ seg.}}{740}$$

Finalmente $T = 1,35135$ micro-segundos (hoy estamos en los nanosegundos y frecuencias de GigaHertz)

- Usa Arquitectura Harvard, es decir, almacenamiento separado de programas y datos. Contrario a la mayoría de los diseños con arquitectura de Harvard, que utilizan buses separados, el 4004, con su necesidad de mantener baja la cuenta de pines, usaba un bus de 4 bits multiplexado para transferir:

Un bus multiplexado es aquel que utiliza un mismo bus para llevar señales distintas, en este caso y debido a que se usan en momentos distintos, el bus de 4 bits se usa para llevar direcciones pero también para llevar un dato.

- 12 bits de direcciones (direccionando hasta 4 KB)

En este caso el bus de direcciones es un bus virtual pues el bus real es de 4 bits, de todas maneras e invirtiendo mucho tiempo, se tiene que la capacidad de direccionamiento es de

$K = 2^{12}$ direcciones distintas, como cada dirección lo hace a un byte, haciendo las cuentas de la Lectura 5 da los 4 KBy de memoria principal.

- El conjunto de instrucciones está formado por 46 instrucciones (de las cuales 41 son de 8 bits de ancho y 5 de 16 bits de ancho)

En la página www.alpern.com.ar/4004.HTM se puede encontrar el repertorio completo de instrucciones del 4004.

Un detalle importante es que el mismo posee instrucciones de 1 byte y de 2 byte, es decir que este microprocesador es un uP CICS, esto significa “cómputo de conjunto de instrucciones complejas” (Complex Instruction Set Computing). Este tipo de uP reinaron hasta la aparición de los Pentium que cambian al concepto de “cómputo con conjunto de instrucciones reducidas” (Reduced Instruction Set Computing – RISC)

Cabe señalar que el 4004 y todos los uP son de una estructura denominada CPU pues reúnen en su interiores ALU y la Unidad de Control de una Computadora.

- 16 registros de 4 bits cada uno

Nótese que la PetiComputadora tenía un solo registro interno a la unidad de control y, en este caso, se tienen 16 registros.-

1.2.- La historia continúa y se cumple el precepto “Creced y multiplicaos”

Luego del éxito del Intel 4004, que no fue tal para Busicom pues la misma quebró, pero Intel se expandió y sacó el Intel 4040 que trabajaba con un mayor grado de integración que el modelo anterior y luego el Intel 8008 que es un uP de 8 bits.

Pero también aparecieron en el mercado otros fabricantes, tal como Rockwell Internacional con su 4-bit PPS-4, Nacional Semiconductor 16-bit IMP-16 , estos micros conjuntamente con los anteriormente nombrados de Intel constituyen lo que se denomina la **Primera Generación** de Microprocesadores, todos aparecen entre 1971 y 1972.

En 1973, aparece la **Segunda generación**, que está tipificada por el Motorola M6800 y el Intel 8080 de 8 bits ambos pero distintos de la Primera Generación por los avances que se producen en la tecnología de los semiconductores para fabricar los chips. La nueva tecnología incrementa la velocidad de procesamiento de las instrucciones y permite una mayor cantidad de transistores en el mismo tamaño de chip. Dentro de esta misma generación se tiene el Fairchild F-8 de 8-bits, el Texas Instruments SBP de 4-bits, el General Instrument CP-1600 de 16-bit (y poco éxito) y el Signetics 2650 de 8 bits. Signetics es una empresa norteamericana, subsidiaria de Philips de Holanda y será motivo de un estudio detallado en el siguiente punto, pues conceptualmente presenta una estructura lógica de sumo interés, así como un set de instrucciones sumamente completo en distintos tipos de direccionamiento.

En 1976 aparece la **generación dos y medio** cuyo referente emblemático es el Zilog Z-80 de 8-bits. En 1978 aparece la **Tercera Generación** de Microprocesadores que da origen a la **Primera Generación de Computadoras Personales (PC)** que constituye el comienzo de la Computación Socializada y de la expansión de la Informática a partir de 1981. Los referentes más importantes son el Zilog Z-8000 de 16-bits, el Intel 8086 también de 16-bits y el Intel 8088 de 16-bits.

1.3.- Microprocesador Signetics 2650

Haremos un estudio detallado del Signetics 2650, en lo que respecta a la Arquitectura de las Instrucciones, estudio que se puede generalizar conceptualmente para todos los microprocesadores de tipo CPU presentes en el mercado.

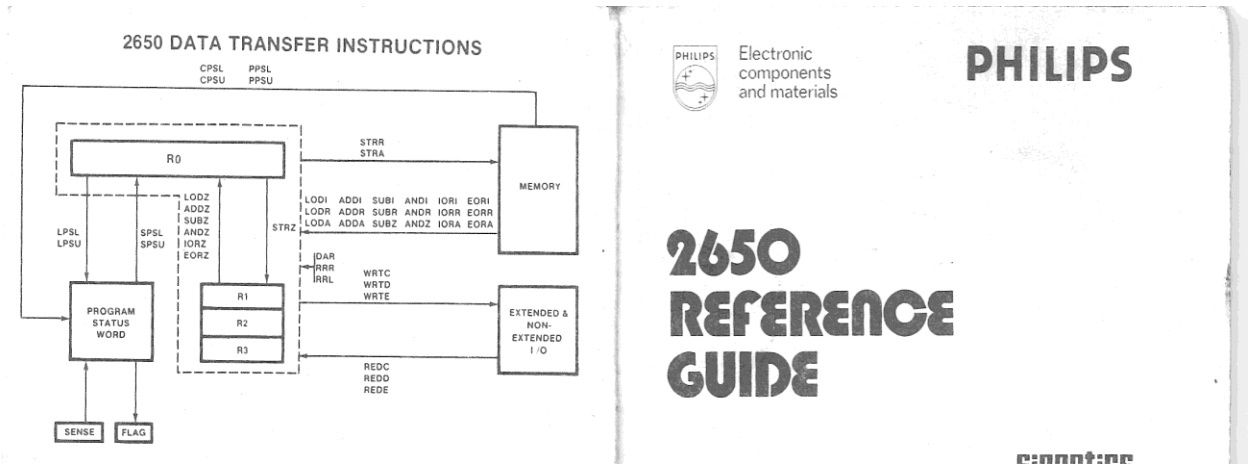


Figura 3 : Set de Instrucciones del Signetics 2650 con indicación de las instrucciones que vinculan los registros de la CPU con la Memoria y las puertas de entrada/salida

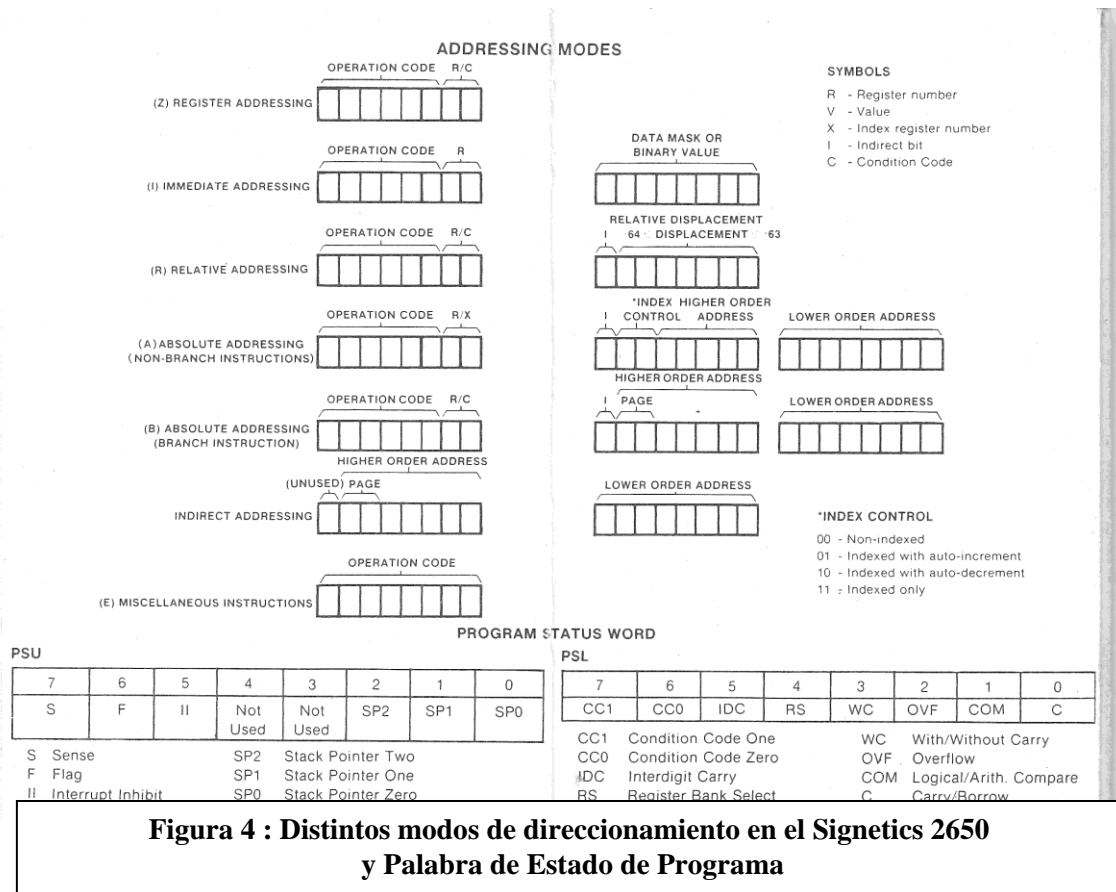


Figura 4 : Distintos modos de direccionamiento en el Signetics 2650 y Palabra de Estado de Programa

En la figura 3 se puede observar la arquitectura básica del 2650, al respecto sus características técnicas más importantes son:

- Microprocesador de 8 bits
- Máxima velocidad del reloj 3,58 MHz

- Usa Arquitectura Harvard, es decir, almacenamiento separado de programas y datos. Utilizan buses separados de direccionamiento y de datos.
- 15 bits de direcciones (direccionando hasta 32 KB)
- 7 Registros de Uso General, Uno tomado como Registro Acumulador y seis divididos en dos bancos de 3(tres) Registros cada uno.
- Presenta una PSW de 16 bits. La Palabra de Estado de Programa es un registro de la Unidad de Control cuyos bits se denominan flags(banderas) y sirven para tomar decisiones.
- El Registro de Instrucción tiene una estructura de Tres Palabra Lógicas. La primera palabra lógica es el Registro de Instrucción propiamente dicho pues allí se encuentra el código de operación, en las siguientes palabra lógicas llamadas Holding Register Low y Holding Register High, se encontrarán datos o direcciones.
- En la Unidad de Control se tiene un sumador de Direcciones (Adder Address) a fin de obtener direccionamientos de datos en forma calculada o indirecta.

En la Figura 4 se pueden ver los modos de Direccionamiento que son:

Direccionamiento Inherente o Register Addressing: Tiene un formato de Una Palabra Lógica(8 bits).

El Ordenamiento de los campos indica que existen 6 bits para el Código de Operación propiamente dicho y 2(dos) bits que indican el Registro con el cuál Opera el Registro Cero o Registro Acumulador.

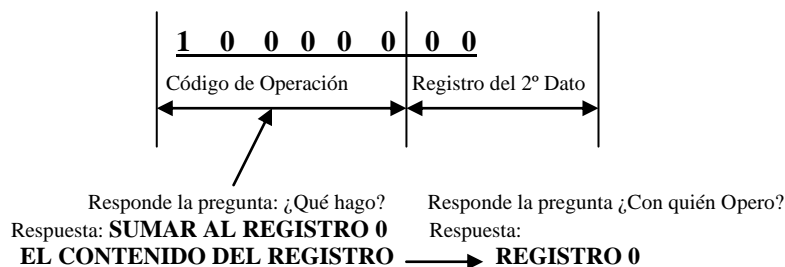
Trabaja con el concepto Acumulador-Registro, es decir, el primer operando está en el R_0 y el segundo Operando se encuentra en uno de los registros o en el mismo Registro cero.

En el ciclo de instrucción se usan solamente dos pulsos de reloj y se utiliza unicamente la Primera Palabra Lógica del IR.-

Se tratará unicamente la instrucción de suma en todos los casos de direccionamiento. En el modo de Direccionamiento Inherente y con referencia a la Figura 5, se trabajará con la instrucción ADDZ.

La misma tiene cuatro versiones en Hexadecimal: 80, 81, 82 y 83.

El formato de $(80)_{16}$ será:



Esto implica que si se tiene ADDZ (80) el sentido semántico es:

$$(R_0) + (R_0) \longrightarrow (R_0)$$

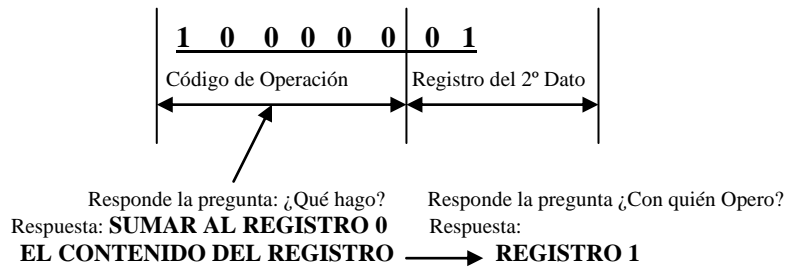
Si antes de la ejecución de la instrucción se tiene: $(R_0) = 2$

Después de la ejecución de la Instrucción se tiene: $(R_0) = 4$

Se duplica el contenido del R_0 o, dicho de otra manera, se multiplica por dos el contenido del registro.

Cabe señalar que la suma se hace con el ALU.

El formato de $(81)_{16}$ será:



Esto implica que si se tiene ADDZ (81) el sentido semántico es:

$$(R_0) + (R_1) \longrightarrow (R_0)$$

Pero es necesario saber si es el R_1 del Banco 0 o del Banco 1, para dilucidar esta duda, se debe mirar en la Palabra de Estado de Programa (PSW) que en otros micros se llama Registro del Código de Condición (CCR). En la Figura 4 se puede observar que el bit 4 de la PSL (parte baja de la PSW) es el Selector de Banco. Supongamos que $RS = 1$ (Se trabaja con el Banco 1)

Si antes de la ejecución de la instrucción se tiene: $(R_0) = 2$ (R_1) del Banco 0 es 3 y el R_1) del Banco 1 es 5.

Después de la ejecución de la Instrucción se tiene: $(R_0) = 7$

Se suma al Registro Cero el contenido del Registro Uno del Banco 1. El Registro Uno no cambia su contenido.

Cabe señalar que la suma se hace en el ALU

Y de esta manera se opera con las otras instrucciones de este grupo.

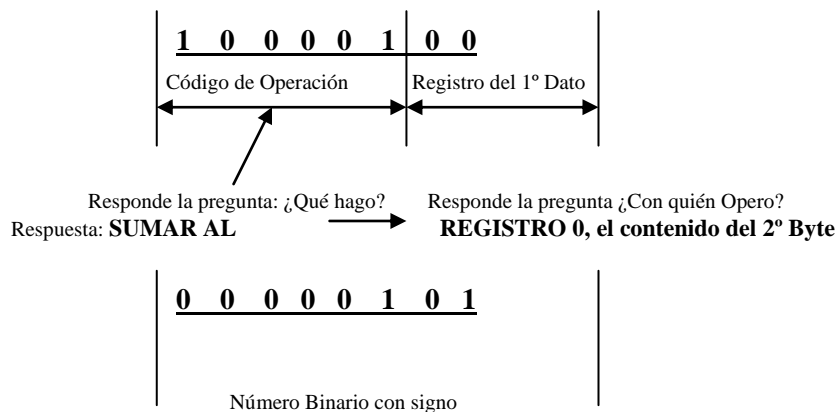
Direccionamiento Inmediato o Immediate Addressing: Tiene un formato de Dos Palabras Lógicas (16 bits). Este tipo de instrucciones tienen la característica que el segundo dato se encuentra directamente en el formato de la instrucción, en contraposición de las otras instrucciones que hacen referencia a los datos indicándolos como el contenido de una posición de memoria de la Memoria Principal.

En este tipo de instrucciones se usan cuatro pulsos de reloj y la Primera y Segunda Palabra del Registro de Instrucción, es decir, el Registro de Instrucción en sí y el Holding Register High.

Se tratará únicamente la instrucción de suma como en todos los casos de direccionamiento. En el modo de Direccionamiento Inmediato y con referencia a la Figura 5, se trabajará con la instrucción ADDI.

La misma tiene cuatro versiones en Hexadecimal: 84, 85, 86 y 87.

El formato de $(84)_{16}$ será:



Esto implica que si se tiene ADDZ (84) el sentido semántico es:

$$(R_0) + (2^{\circ} \text{ Byte de la Instrucción}) \longrightarrow (R_0)$$

Si antes de la ejecución de la instrucción se tiene: $(R_0) = 2$

Después de la ejecución de la Instrucción se tiene: $(R_0) = 7$

El ciclo de la Instrucción será más lento que el ciclo de las instrucciones de Direcccionamiento Inherente pues se necesita llevar al IR dos bytes, cada uno de ellos requiere de dos pulsos de reloj.

Cabe señalar que la suma se hace con el ALU.

	DESCRIPTION OF OPERATION	FORMAT	MNE-MONIC	OP CODE R or CC				BYTES	CYCLES	BIT FOR-MAT	PSW BITS AFFECTED							NOTE	
				0	1	2	3				CC	IDC	C	OVF	SP	II	F		
LOAD/STORE	Load register zero	2	LOD	Z	00	01	02	03	1	2	Z	•							1
	Load immediate	5		I	04	05	06	07	2	2	I	•							1
	Load relative	7		R	08	09	0A	0B	2	3	R	•							1.6
	Load absolute	10		A	0C	0D	0E	0F	3	4	A	•							6
	Store register zero	2	STR	Z	—	C1	C2	C3	1	2	Z	•							1
	Store relative	7		R	C8	C9	CA	CB	2	3	R	•							6
Store absolute	10	A		CC	CD	CE	CF	3	4	A	•							6	
ARITHMETIC	Add to register zero w/wo carry	2	ADD	Z	80	81	82	83	1	2	Z	•	•	•	•				1
	Add immediate w/wo carry	5		I	84	85	86	87	2	2	I	•	•	•	•				1
	Add relative w/wo carry	7		R	88	89	8A	8B	2	3	R	•	•	•	•				1.6
	Add absolute w/wo carry	10		A	8C	8D	8E	8F	3	4	A	•	•	•	•				1.6
	Subtract from register zero w/wo borrow	2	SUB	Z	A0	A1	A2	A3	1	2	Z	•	•	•	•				1
	Subtract immediate w/wo borrow	5		I	A4	A5	A6	A7	2	2	I	•	•	•	•				1
	Subtract relative w/wo borrow	7		R	A8	A9	AA	AB	2	3	R	•	•	•	•				1.6
	Subtract absolute w/wo borrow	10	A	AC	AD	AE	AF	3	4	A	•	•	•	•				1.6	
	Decimal adjust register	3	DAR		94	95	96	97	1	3	Z	•							1.10
	LOGICAL	AND to register zero	2	AND	Z	—	41	42	43	1	2	Z	•						
AND immediate		5	I		44	45	46	47	2	2	I	•							1
AND relative		7	R		48	49	4A	4B	2	3	R	•							1.6
AND absolute		10	A		4C	4D	4E	4F	3	4	A	•							1.6
Inclusive-OR to register zero		2	IOR	Z	60	61	62	63	1	2	Z	•							1
Inclusive-OR immediate		5		I	64	65	66	67	2	2	I	•							1
Inclusive-OR relative		7		R	68	69	6A	6B	2	3	R	•							1.6
Inclusive-OR absolute		10	A	6C	6D	6E	6F	3	4	A	•							1.6	
Exclusive-OR to register zero		2	EOR	Z	20	21	22	23	1	2	Z	•							1
Exclusive-OR immediate		5		I	24	25	26	27	2	2	I	•							1
Exclusive-OR relative	7	R		28	29	2A	2B	2	3	R	•							1.6	
Exclusive-OR absolute	10	A		2C	2D	2E	2F	3	4	A	•							1.6	
ROTATE/COMPARE	Compare to register zero arithmetic/logical	2	COM	Z	E0	E1	E2	E3	1	2	Z	•							2
	Compare immediate arithmetic/logical	5		I	E4	E5	E6	E7	2	2	I	•							3
	Compare relative arithmetic/logical	7		R	E8	E9	EA	EB	2	3	R	•							3.6
	Compare absolute arithmetic/logical	10		A	EC	ED	EE	EF	3	4	A	•							3.6
	Rotate register w/wo carry	3	RRR		50	51	52	53	1	2	Z	•	•	•	•				1
	Rotate register left w/wo carry	3	RRL		D0	D1	D2	D3	1	2	Z	•	•	•	•				1
BRANCH	Branch on condition true relative	7	BCT	R	18	19	1A	1B	2	3	R								7.8
	Branch on condition true absolute	8		A	1C	1D	1E	1F	3	3	B								7.8
	Branch on condition false relative	7	BCF	R	98	99	9A	—	2	3	R								7
	Branch on condition false absolute	8		A	9C	9D	9E	—	3	3	B								7
	Branch on register non-zero relative	7	BRN	R	58	59	5A	5B	2	3	R								7.8
	Branch on register non-zero absolute	8		A	5C	5D	5E	5F	3	3	B								7.8

Figura 5 : Parte del repertorio de Instrucciones del Signetics 2650

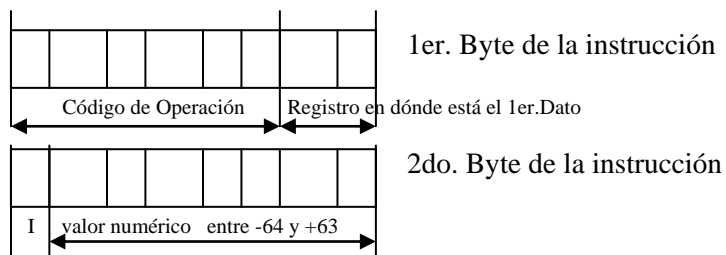
	DESCRIPTION OF OPERATION	FORM MAT	MNE- MONIC	OP CODE R or CC				BIT FOR- MAT	PSW BITS AFFECTED							NOTE								
				0	1	2	3		BYTES	CYCLES	CC	IDC	C	OVF	SP		II	F						
BRANCH	Branch on incrementing register relative	7	BIR {	R	D8	D9	DA	DB	2	3	R									7.8				
	Branch on incrementing register absolute	8		A	DC	DD	DE	DF	3	3	B										7.8			
	Branch on decrementing register relative	7	BDR {	R	F8	F9	FA	FB	2	3	R									7.8				
	Branch on decrementing register absolute	8		A	FC	FD	FE	FF	3	3	B										7.8			
	Zero branch relative, unconditional	6	ZBRR					9B	2	3	ER										6			
	Branch indexed absolute, unconditional	9	BXA					9F	3	3	EB										5.6			
SUBROUTINE BRANCH/RETURN	Branch to subroutine on condition true, relative	7	BST {	R	38	39	3A	3B	2	3	R										7.8			
	Branch to subroutine on condition true, absolute	8		A	3C	3D	3E	3F	3	3	B											7.8		
	Branch to subroutine on condition false, relative	7	BSF {	R	B8	B9	BA	—	2	3	R										7			
	Branch to subroutine on condition false, absolute	8		A	BC	BD	BE	—	3	3	B											7		
	Branch to subroutine on non-zero register, relative	7	BSN {	R	78	79	7A	7B	2	3	R											7.8		
	Branch to subroutine on non-zero register, absolute	8		A	7C	7D	7E	7F	3	3	B											7.8		
	Zero branch to subroutine relative, unconditional	6	ZBSR					BB	2	3	ER											6		
	Branch to subroutine, indexed, absolute unconditional	9	BSXA					BF	3	3	EB											5.6		
	Return from subroutine, conditional	3	RET {	C	14	15	16	17	1	3	Z											8		
	Return from subroutine and enable interrupt, conditional	3		E	34	35	36	37	1	3	Z												8	
	INPUT/ OUTPUT	Write data	3	WRD		F0	F1	F2	F3	1	2	Z												
Read data		3	REDD		70	71	72	73	1	2	Z												1	
Write control		3	WRTC		B0	B1	B2	B3	1	2	Z													
Read control		3	REDC		30	31	32	33	1	2	Z													
Write extended		5	WRTE		D4	D5	D6	D7	2	3	I													
Read extended		5	REDE		54	55	56	57	2	3	I													1
MISC.	Halt, enter wait state	1	HALT		40	—	—	—	1	1	E													
	No operation	1	NOP		C0	—	—	—	1	1	E													
	Test under mask immediate	5	TMI		F4	F5	F6	F7	2	3	I												4	
PROGRAM STATUS	Load program status, upper	1	LPS {	U	—	—	92	—	1	2	E													
	Load program status, lower	1		L	—	—	93	—	1	2	E													
	Store program status, upper	1	SPS {	U	—	—	12	—	1	2	E													
	Store program status, lower	1		L	—	—	13	—	1	2	E													
	Clear program status, upper, masked	4	CPS {	U	—	—	74	—	2	3	EI													
	Clear program status, lower, masked	4		L	—	—	75	—	2	3	EI													
	Preset program status, upper, masked	4	PPS {	U	—	—	76	—	2	3	EI													
	Preset program status, lower, masked	4		L	—	—	77	—	2	3	EI													
	Test program status, upper, masked	4	TPS {	U	—	—	B4	—	2	3	EI													
	Test program status, lower, masked	4		L	—	—	B5	—	2	3	EI													

Figura 6 : Segunda parte del repertorio de instrucciones del Signetics 2650

Direccionamiento Relativo o Relative Addressing:

Tiene un formato de Dos Palabras Lógicas (16 bits). Este tipo de instrucciones tienen la característica que la dirección del segundo dato debe calcularse utilizando el valor numérico con signo que se encuentra en el 2º Byte de la Instrucción.

El formato de la instrucción es el siguiente:



El direccionamiento se denomina Relativo y habrá dos maneras distintas de encontrar la Dirección del 2º Dato, a saber,

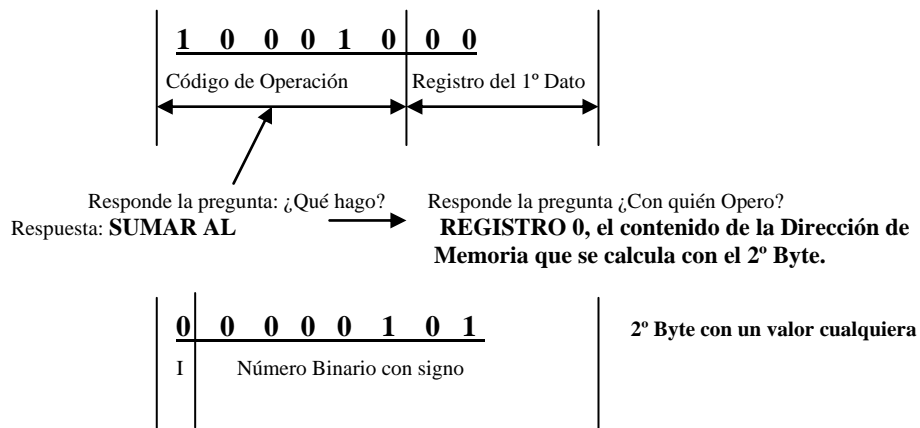
- Si $I = 0$, se está en presencia de un Direccionamiento Relativo referido al Contador de Programa.
- Si $I = 1$, se está en presencia de un Direccionamiento Relativo referido al Registro Índice.

En este tipo de instrucciones se usan cuatro pulsos de reloj y la Primera y Segunda Palabra del Registro de Instrucción, es decir, el Registro de Instrucción en sí y el Holding Register High

Se tratará únicamente la instrucción de suma como en todos los casos de direccionamiento. En el modo de Direccionamiento Relativo y con referencia a la Figura 5, se trabajará con la instrucción ADDR.

La misma tiene cuatro versiones en Hexadecimal: 88, 89, 8A y 8B.

El formato de $(88)_{16}$ será:



Vamos a tratar de mostrar el modo de cálculo de la Dirección del 2º Dato mediante la corrida del siguiente programa:

MEM	CONTENIDO	Registros	Contenido
3487	ADDR $(88)_{16}$	PC	3487
3488	0 2	R ₀	02
3489	0 3	R ₁	A1
348A	0 0	R ₂	12
348B	0 1	R ₃	13
348C	0 5	R ₁	B1
348D	1 1	R ₂	13
348E	1 2	R ₃	15
		XR	348B

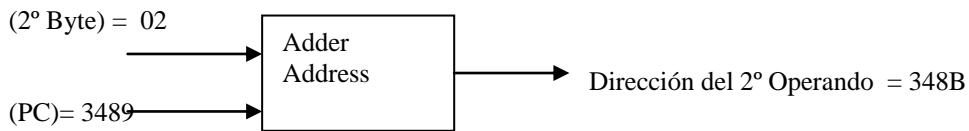
Corrida del Programa con I= 0

Memoria 3487

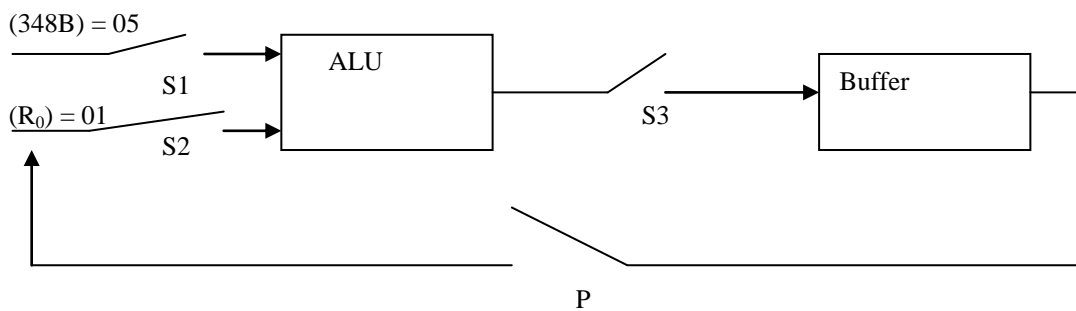
- 1.- $(3487) \rightarrow (IR)$
 $(88)_{16} = (IR)$
- 2.- $(PC) + 1 \rightarrow (PC)$
 $3488 = (PC)$
- 3.- Interpretación : La máquina se da cuenta que debe inhibir la Ejecución pues la instrucción no está completa.
- 4.- La Ejecución se inhibe debido a que falta parte de la instrucción.

Memoria 3488

- 1.- $(3488) \rightarrow (HR)_{HIGH}$
 $(02)_{16} = (HR)_{HIGH}$
- 2.- $(PC) + 1 \rightarrow (PC)$
 $3489 = (PC)$
- 3.- Interpretación : La máquina se da cuenta que debe trabajar para obtener la Dirección del 2º Dato. Sabe que es relativa al Contador de Programa pues $I = 0$. Para el cálculo debe tomar el valor numérico del 2º Byte y el valor actual de PC que es 348A, usará el Adder Address que se halla en la Unidad de Control



La Interpretación continúa preparando la suma que indica la instrucción (ADDR) mediante el ALU:



Con S1, S2 y S3 cerrados y P abierto, el resultado de 06 aparece en el Buffer.

4.- La Ejecución se produce cuando S1, S2 y S3 se abren y P se cierra y se abre, el contenido del Buffer pasa al Registro Cero y la instrucción concluye. Finalmente $(R_0) = 07$

Veamos la corrida del siguiente programa

MEM	CONTENIDO	Registros	Contenido
3487	ADDR $(88)_{16}$	PC	3487
3488	8 2	R ₀	02
3489	0 3	R ₁	A1
348A	0 0	R ₂	12
348B	0 1	R ₃	13
348C	0 5	R ₁	B1
348D	1 1	R ₂	13
348E	1 2	R ₃	15
		XR	348B

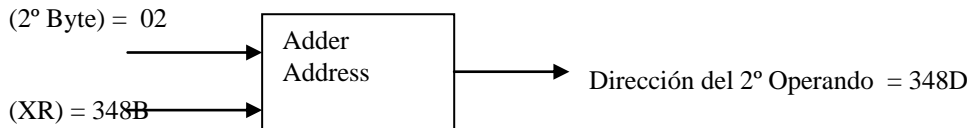
Corrida del Programa con $I = 1$

Memoria 3487

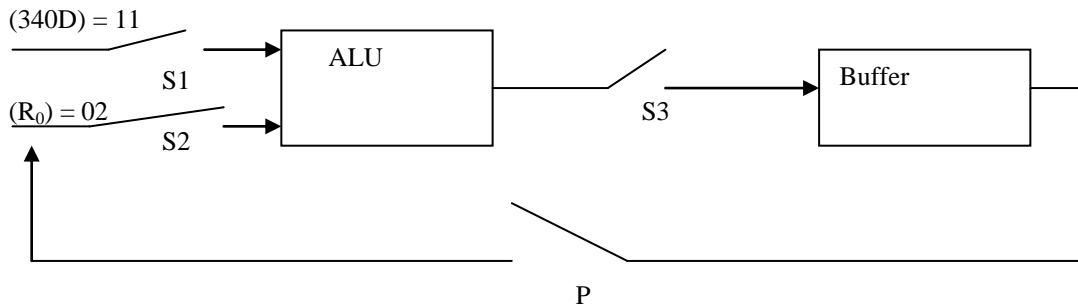
- 1.- $(3487) \rightarrow (IR)$
 $(88)_{16} = (IR)$
- 2.- $(PC) + 1 \rightarrow (PC)$
 $3488 = (PC)$
- 3.- Interpretación : La máquina se da cuenta que debe inhibir la Ejecución pues la instrucción no está completa.
- 4.- La Ejecución se inhibe debido a que falta parte de la instrucción.

Memoria 3488

- 1.- $(3488) \rightarrow (HR)_{HIGH}$
 $(82)_{16} = (HR)_{HIGH}$
- 2.- $(PC) + 1 \rightarrow (PC)$
 $3489 = (PC)$
- 3.- Interpretación : La máquina se da cuenta que debe trabajar para obtener la Dirección del 2º Dato. Sabe que es relativa al Registro índice (XR) pues I= 1. Para el cálculo debe tomar el valor numérico de los últimos 7 bits del 2º Byte y el valor del (XR) que es 348B, usará el Adder Address que se halla en la Unidad de Control



La Interpretación continúa preparando la suma que indica la instrucción (ADDR) mediante el ALU:



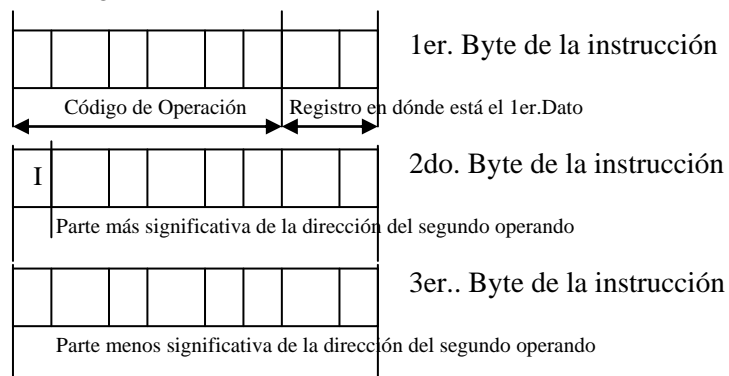
Con S1, S2 y S3 cerrados y P abierto, el resultado de 13 aparece en el Buffer.

- 4.- La Ejecución se produce cuando S1,S2 y S3 se abren y P se cierra y se abre, el contenido del Buffer pasa al Registro Cero y la instrucción concluye. Finalmente $(R_0) = 13$.

Direccionamiento Absoluto o Absolute Addressing:

Tiene un formato de Tres Palabras Lógicas (24 bits). Este tipo de instrucciones tienen la característica que se direcciona en forma directa la posición de memoria del segundo operando.

El formato de la instrucción es el siguiente:



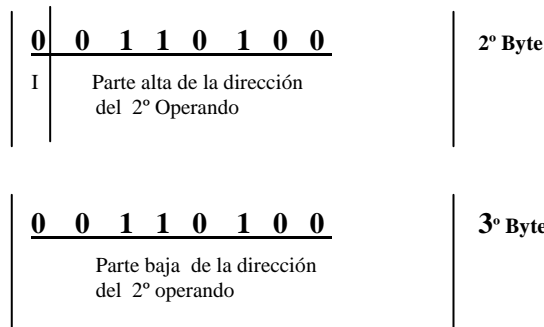
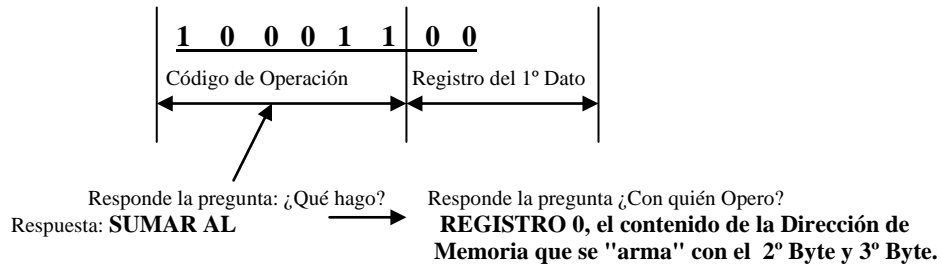
El direccionamiento requiere de 15 bits pues si bien el PC es una doble palabra lógica (16 bits) solamente se utilizan en direccionar 15 bits, por eso que la memoria principal se limita a 32 KBy.

En este tipo de instrucciones se usan seis pulsos de reloj y la totalidad del Registro de Instrucción, es decir, el Registro de Instrucción en sí, el Holding Register High y el Holding Register Low.

Se tratará únicamente la instrucción de suma como en todos los casos de direccionamiento. En el modo de Direccionamiento Absoluto y con referencia a la Figura 5, se trabajará con la instrucción ADDA.

La misma tiene cuatro versiones en Hexadecimal: 8C, 8D, 8E y 8F.

El formato de $(8C)_{16}$ será:



Vamos a tratar de mostrar el modo de cálculo de la Dirección del 2º Dato mediante la corrida del siguiente programa:

MEM	CONTENIDO	Registros	Contenido
3487	ADDA $(8C)_{16}$	PC	3487
3488	3 4	R ₀	02
3489	8 E	R ₁	A1
348A	0 0	R ₂	12
348B	0 1	R ₃	13
348C	0 5	R ₁	B1
348D	1 1	R ₂	13
348E	1 2	R ₃	15
		XR	348B

} Banco 0
 } Banco 1

Corrida del Programa

Memoria 3487

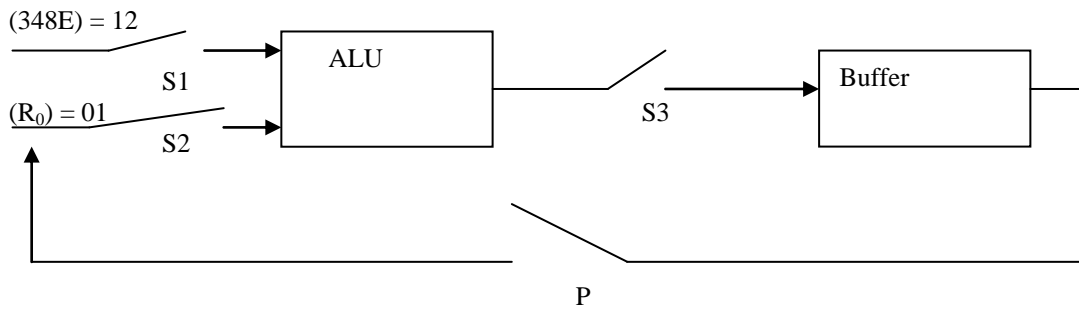
- 1.- $(3487) \rightarrow (IR)$
 $(8C)_{16} = (IR)$
- 2.- $(PC) + 1 \rightarrow (PC)$
 $3488 = (PC)$
- 3.- Interpretación : La máquina se da cuenta que debe inhibir la Ejecución pues la instrucción no está completa.
- 4.- La Ejecución se inhibe debido a que falta parte de la instrucción.

Memoria 3488

- 1.- $(3488) \longrightarrow (HR)_{HIGH}$
 $(34)_{16} = (HR)_{HIGH}$
- 2.- $(PC) + 1 \longrightarrow (PC)$
 $3489 = (PC)$
- 3.- Interpretación : La máquina se da cuenta que debe volver a inhibir la ejecución pues aún falta parte de la instrucción.
- 4.- La Ejecución se inhibe debido a que falta parte de la instrucción.

Memoria 3489

- 1.- $(3489) \longrightarrow (HR)_{LOW}$
 $(8E)_{16} = (HR)_{LOW}$
- 2.- $(PC) + 1 \longrightarrow (PC)$
 $348A = (PC)$
- 3.- Interpretación : La máquina realiz ala operación y determina que debe sumar el Contenido del Registro 0 con el contenido de la Memoria 348E y coloca el resultado en el Buffer.



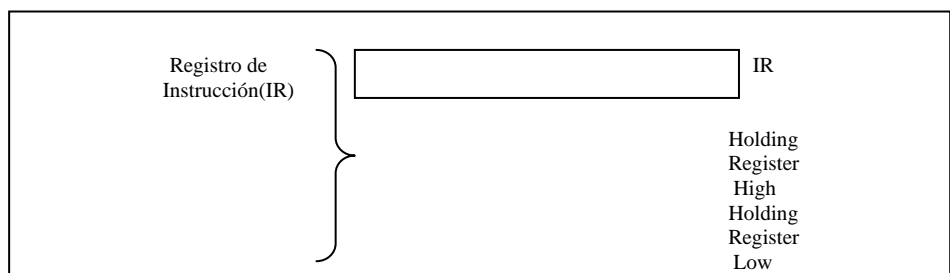
Con S1, S2 y S3 cerrados y P abierto, el resultado de 13 aparece en el Buffer.

- 4.- La Ejecución se produce cuando S1,S2 y S3 se abren y P se cierra y se abre, el contenido del Buffer pasa al Registro Cero y la instrucción concluye. Finalmente $(R_0) = 13$.

Una última vista del microprocesador Signetics 2650

La Arquitectura de las instrucciones del 2650 no será tratada en profundidad pues eso queda reservado para el Intel 8085 que será visto en la parte de Técnicas Digitales.

La arquitectura de la Unidad de Control del 2650 se puede esquematizar acorde a la Figura 7.



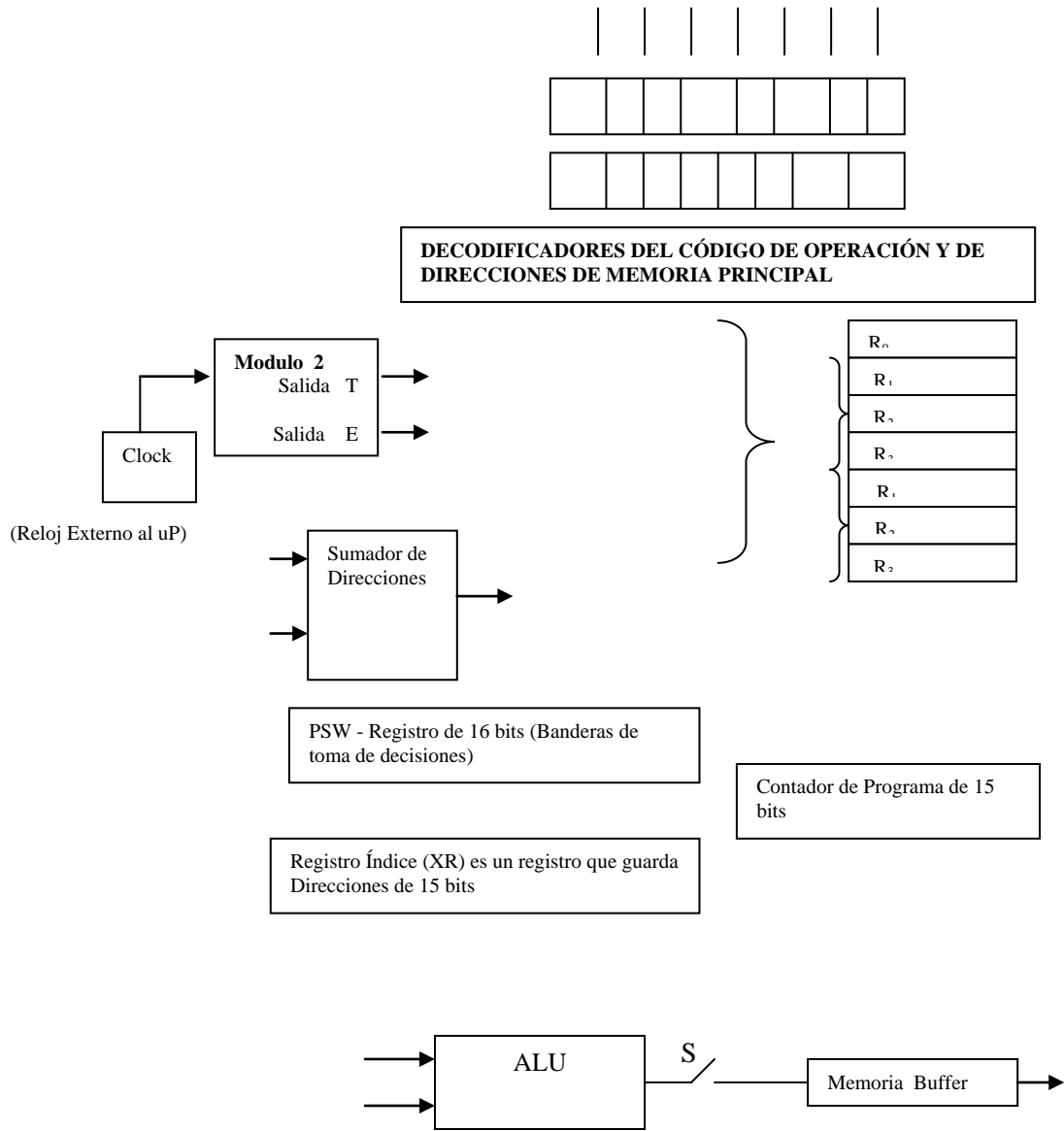


Figura 7: Estructura de esquematizada del 2650