Microprogram Sequencers

2909/2911

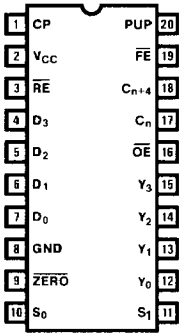
PRELIMINARY DATA

Features/Benefits

- 4-bit slice cascadable to any number of microwords
- Internal address register
- Branch input for N-way branches
- Cascadable 4-bit microprogram counter
- 4 x 4 file with stack pointer and push pop control for nesting microsubroutines
- Zero input for returning to the zero microcode word
- Individual OR input for each bit for branching to higher microinstructions (2909 only)
- Three-state outputs
- All internal registers change state on the LOW-to-HIGH transition of the clock
- 2909 in 28-pin package
- 2911 in 20-pin package

PART NUMBER		PACKAGE		TEMPERATURE RANGE
2909	2911	2909	2911	
2909NC	2911NC	N28	N20	0°C to +70°C
2909JC	2911JC	J28	J20	0°C to +70°C
2909JM	2911JM	J28	J20	−55°C to +125°C
2909FM	—	F28	—	−55°C to +125°C

Pin Configuration

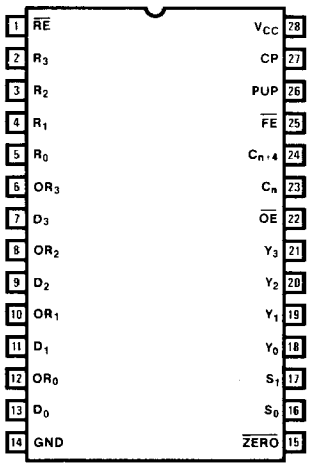


Description

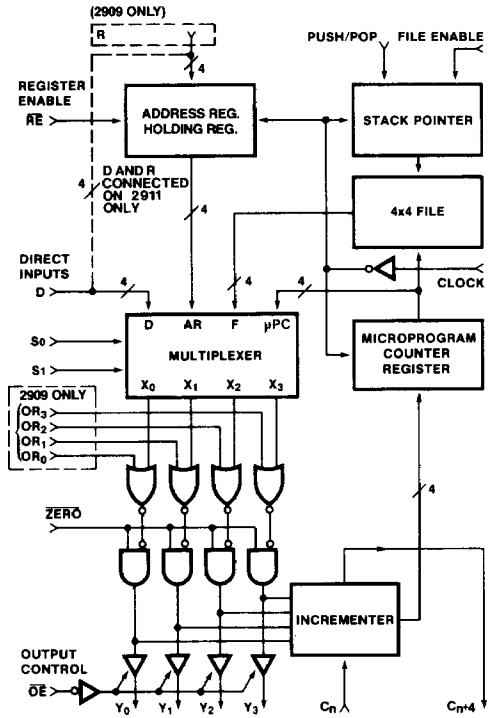
The 2909 is a four-bit wide address controller intended for sequencing through a series of microinstructions contained in a ROM or PROM. Two 2909's may be interconnected to generate an eight-bit address (256 words), and three may be used to generate a twelve-bit address (4K words).

The 2909 can select an address from any of four sources. They are: 1) a set of external direct inputs (D); 2) external data from the R inputs, stored in an internal register; 3) a four-word deep push/pop stack; or 4) a program counter register (which usually contains the last address plus one). The push/pop stack includes certain control lines so that it can efficiently execute nested subroutine linkages. Each of the four outputs can be OR'ed with an external input for conditional skip or branch instructions, and a separate line forces the outputs to all zeroes. The outputs are three-state.

The 2911 is an identical circuit to the 2909, except the four OR inputs are removed and the D and R inputs are tied together. The 2911 is in a 20-pin, 0.3″ centers package.



Block Diagram



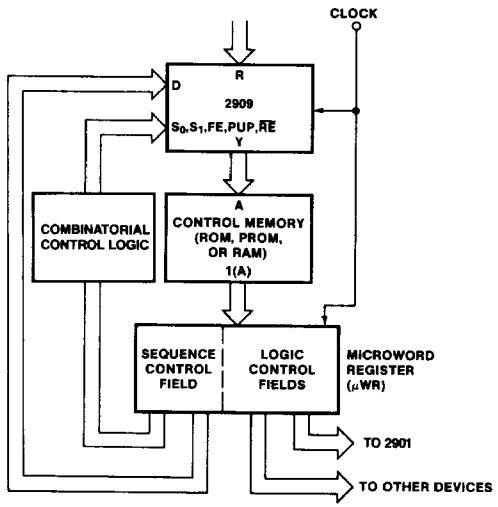


Figure 3. Microprogram Sequencer Control.

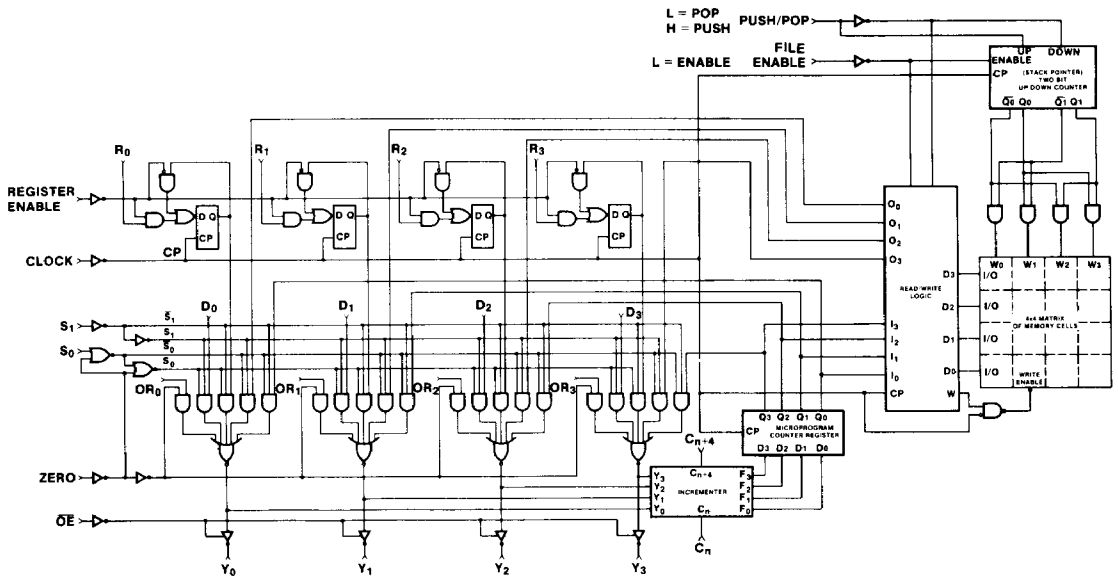


Figure 2. Microprogram Sequencer Block Diagram.

Absolute Maximum Ratings

Storage temperature ..... −65°C to +150°C
Temperature (ambient) under bias ..... −55°C to +125°C
Supply voltage to ground potential ..... −0.5 V to +7.0 V
DC voltage applied to outputs for HIGH output state ..... −0.5 V to +V_{CC} max.
DC input voltage ..... −0.5 V to +7.0 V
DC output current, into outputs ..... 30 mA
DC input current ..... −30 mA to +5.0 mA

Electrical Characteristics

Over Recommended Operating Range

SYMBOL	PARAMETER	TEST CONDITIONS[1]		MILITARY MIN	MILITARY TYP	MILITARY MAX	COMMERCIAL MIN	COMMERCIAL TYP	COMMERCIAL MAX	UNIT
V_{OH}	Output HIGH voltage	V_{CC} = MIN., V_{IN} = V_{IH} or V_{IL}	I_{OH} = −1.0mA	2.4						V
			I_{OH} = −2.6mA				2.4			
V_{OL}	Output LOW voltage	V_{CC} = MIN., V_{IN} = V_{IH} or V_{IL}	I_{OL} = 4.0mA			0.4			0.4	V
			I_{OL} = 8.0mA			0.45			0.45	
			I_{OL} = 12mA[5]			0.5			0.5	
V_{IH}	Input HIGH level	Guaranteed input logical HIGH voltage for all inputs		2.0			2.0			V
V_{IL}	Input LOW level	Guaranteed input logical LOW voltage for all inputs				0.7			0.8	V
V_I	Input clamp voltage	V_{CC} = MIN., I_{IN} = −18mA				−1.5			−1.5	V
I_{IL}	Input LOW current	V_{CC} = MAX, V_{IN} = 0.4V	C_n			−1.08			−1.08	mA
			Push/Pop, $\overline{OE}$			−0.72			−0.72	
			Others[6]			−0.36			−0.36	
I_{IH}	Input HIGH current	V_{CC} = MAX., V_{IN} = 2.7V	C_n			40			40	μA
			Push/Pop			40			40	
			Others[6]			20			20	
I_I	Input HIGH current	V_{CC} = MAX., V_{IN} = 7.0V	C_n, Push/Pop			0.2			0.2	mA
			Others[6]			0.1			0.1	
I_{OS}	Output short circuit current[3]	V_{CC} = MAX	Y_0−Y_3	−30		−100	−30		−100	mA
			C_n+4	−30		−85	−30		−85	
I_{CC}	Power supply current	V_{CC} = MAX[4]			80	130		80	130	mA
I_{OZL}	Output OFF current	V_{CC} = MAX., $\overline{OE}$ = 2.7V	V_{OUT} = 0.4V			−20			−20	μA
I_{OZH}			V_{OUT} = 2.7V			20			20	

NOTES:
1. For conditions shown as MIN. or MAX., use the appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. Apply GND to C_n, R_0, R_1, R_2, R_3, OR_0, OR_1, OR_2, OR_3, D_0, D_1, D_2, and D_3. Other inputs open. All outputs open. Measured after a LOW-to-HIGH clock transition.
5. The 12mA guarantee applies only to Y_0, Y_1, Y_2 and Y_3.
6. For the 2911, D_i, and R_i are internally connected. Loading is doubled (to same values as Push/Pop).

Switching Characteristics

Over Operating Range

All parameters are guaranteed worst case over the operating voltage and temperature range for the device type. (Grade C = 0°C to +70°C, 4.75V to 5.25V; Grade M = −55°C to +125°C, 4.5V to 5.5V)

Table 1
Minimum Clock Requirements

Minimum Clock LOW Time	50
Minimum Clock HIGH Time	30

Table 2
Maximum Combinatorial Propagation Delays

INPUTS \ OUTPUTS	Y_i	C_{n+4}
$\overline{OE}$	25	—
$\overline{ZERO}$	35	45
OR_i	20	32
S_0, S_1	40	50
D_i	20	32
C_n	—	18

Table 3
Maximum Delays
From Clock to Outputs

FUNCTIONAL PATH	GRADE	CLOCK TO Y_i	CLOCK TO C_{n+4}
Register (S_1 S_0 = LH)	C	48	58
	M	55	65
μ Program Counter (S_1 S_0 = LL)	C	48	58
	M	55	65
File (S_1 S_0 = HL)	C	70	80
	M	80	90

R_L = 2.0kΩ C_L = 15pF

Table 4
Set-up and Hold Time
Requirements

EXTERNAL INPUTS	t_s	t_h
$\overline{RE}$	20	5.0
R_i	15	0
PUSH/POP	20	5.0
$\overline{FE}$	20	0
C_n	15	0
D_i	20	0
OR_i	20	0
S_0, S_1	40	0
$\overline{ZERO}$	40	0

Architecture of the 2090/2911

The 2909/2911 are bipolar microprogram sequencers intended for use in high-speed microprocessor applications. The device is a cascadable 4-bit slice such that two devices allow addressing of up to 256 words of microprogram and three devices allow addressing of up to 4K words of microprogram. A detailed logic diagram is shown in Figure 2.

The device contains a four-input multiplexer that is used to select either the address register, direct inputs, microprogram counter, or file as the source of the next microinstruction address. This multiplexer is controlled by the S_0 and S_1 inputs.

The address register consists of four D-type, edge triggered flip-flops with a common clock enable. When the address register enable is LOW, new data is entered into the register on the clock LOW-to-HIGH transition. The address register is available at the multiplexer as a source for the next microinstruction address. The direct input is a four-bit field of inputs to the multiplexer and can be selected as the next microinstruction address. On the 2911, the direct inputs are also used as inputs to the register. This allows an N-way branch where N is any word in the microcode.

The 2909/2911 contains a microprogram counter (μPC) that is composed of a 4-bit incrementer followed by a 4-bit register. The incrementer has carry-in (C_n) and carry-out (C_{n+4}) such that cascading to larger word lengths is straightforward. The μPC can be used in either of two ways. When the least significant carry-in to the incrementer is HIGH, the microprogram register is loaded on the next clock cycle with the current Y output word plus one (Y+1→μPC.) Thus sequential microinstructions can be executed. If this least significant C_n is LOW, the incrementer passes the Y output word unmodified and the microprogram register is loaded with the same Y word on the next clock cycle (Y→μPC). Thus, the same microinstruction can be executed any number of times by using the least significant C_n as the control.

The last source available at the multiplexer input is the 4 x 4 file (stack). The file is used to provide return address linkage when executing microsubroutines. The file contains a built-in stack pointer (SP) which always points to the last file word written. This allows stack reference operations (looping) to be performed without a push or pop.

The stack pointer operates as an up/down counter with separate push/pop and file enable inputs. When the file enable input is LOW and the push/pop input is HIGH, the PUSH operation is enabled. This causes the stack pointer to increment and the file to be written with the required return linkage—the next microinstruction address following the subroutine jump which initiated the PUSH.

If the file enable input is LOW and the push/pop control is LOW, a POP operation occurs. This implies the usage of the return linkage during this cycle and thus a return from subroutine. The next LOW-to-HIGH clock transition causes the stack pointer to decrement. If the file enable is HIGH, no action is taken by the stack pointer regardless of any other input.

The stack pointer linkage is such that any combination of pushes, pops or stack references can be achieved. One microinstruction subroutines can be performed. Since the stack is 4 words deep, up to four microsubroutines can be nested.

The ZERO input is used to force the four outputs to the binary zero state. When the ZERO input is LOW, all Y outputs are LOW regardless of any other inputs (except $\overline{OE}$). Each Y output bit also has a separate OR input such that a conditional logic one can be forced at each Y output. This allows jumping to different microinstructions on programmed conditions.

The 2909/2911 feature three-state Y outputs. These can be particularly useful in military designs requiring external Ground Support Equipment (GSE) to provide automatic checkout of the microprocessor. The internal control can be placed in the high-impedance state, and preprogrammed sequences of microinstructions can be executed via external access to the control ROM/PROM.

Definition of Terms

A set of symbols is used in this data sheet to represent various internal and external registers and signals used with the 2909. Since its principle application is as a controller for a microprogram store, it is necessary to define some signals associated with the microcode itself. Figure 3 illustrates the basic interconnection of 2909, memory, and microinstruction register. The definitions here apply to this architecture.

Inputs to 2909/2911

S_1, S_0	Control lines for address source selection
$\overline{FE}$, PUP	Control lines for push/pop stack
$\overline{RE}$	Enable line for internal address register
OR_i	Logic OR inputs on each address output line
$\overline{ZERO}$	Logic AND input on the output lines
$\overline{OE}$	Output Enable. When $\overline{OE}$ is HIGH, the Y outputs are OFF (high impedance)
C_n	Carry-in to the incrementer
R_i	Inputs to the internal address register
D_i	Direct inputs to the multiplexer
CP	Clock input to the AR and μPC register and Push-Pop stack

Outputs from the 2909/2911

Y_i	Address outputs from 2909. (Address inputs to control memory.)
C_{n+4}	Carry out from the incrementer

Internal Signals

μPC	Contents of the microprogram counter
REG	Contents of the register
STK0-STK3	Contents of the push/pop stack. By definition, the word in the four-by-four file, addressed by the stack pointer is STK0. Conceptually data is pushed into the stack at STK0; a subsequent push moves STK0 to STK1; a pop implies STK3 → STK2 → STK1 → STK0. Physically, only the stack pointer changes when a push or pop is performed. The data does not move. I/O occurs at STK0.
SP	Contents of the stack pointer.

External to the 2909/2911

A	Address to the control memory
I(A)	Instruction in control memory at address A
μWR	Contents of the microword register (at output of control memory). The microword register contains the instruction currently being executed.
T_n	Time period (cycle) n

7

Address Selection

OCTAL	S_1	S_0	SOURCE FOR Y OUTPUTS	SYMBOL
0	L	L	Microprogram Counter	μPc
1	L	H	Register	REG
2	H	L	Push-Pop stack	STK0
3	H	H	Direct inputs	D_i

Output Control

OR_i	$\overline{ZERO}$	$\overline{OE}$	Y_i
X	X	H	Z
X	L	L	L
H	H	L	H
L	H	L	Source selected by S_0 S1

Z = High Impedance

Synchronous Stack Control

$\overline{FE}$	PUP	PUSH-POP STACK CHANGE
H	X	No change
L	H	Increment stack pointer, then push current PC onto STK0
L	L	Pop stack (decrement stack pointer)

H = High
L = Low
X = Don't Care

Figure 5.

CYCLE	S_1, S_0, $\overline{FE}$, PUP	μPC	REG	STK0	STK1	STK2	STK3	Y_{OUT}	COMMENT	PRINCIPLE USE
N	0 0 0 0	J	K	Ra	Rb	Rc	Rd	J	Pop Stack	End
N+1	—	J+1	K	Rb	Rc	Rd	Ra	—		Loop
N	0 0 0 1	J	K	Ra	Rb	Rc	Rd	J	Push μPC	Set-up
N+1	—	J+1	K	J	Ra	Rb	Rc	—		Loop
N	0 0 1 X	J	K	Ra	Rb	Rc	Rd	J	Continue	Continue
N+1	—	J+1	K	Ra	Rb	Rc	Rd	—		
N	0 1 0 0	J	K	Ra	Rb	Rc	Rd	K	Pop Stack;	End
N+1	—	K+1	K	Rb	Rc	Rd	Ra	—	Use AR for Address	Loop
N	0 1 0 1	J	K	Ra	Rb	Rc	Rd	K	Push μPC;	JSR AR
N+1	—	K+1	K	J	Ra	Rb	Rc	—	Jump to Address in AR	
N	0 1 1 X	J	K	Ra	Rb	Rc	Rd	K	Jump to Address in AR	JMP AR
N+1	—	K+1	K	Ra	Rb	Rc	Rd	—		
N	1 0 0 0	J	K	Ra	Rb	Rc	Rd	Ra	Jump to Address in STKO;	RTS
N+1	—	Ra+1	K	Rb	Rc	Rd	Ra	—	Pop Stack	
N	1 0 0 1	J	K	Ra	Rb	Rc	Rd	Ra	Jump to Address in STKO;	
N+1	—	Ra+1	K	J	Ra	Rb	Rc	—	Push μPc	
N	1 0 1 X	J	K	Ra	Rb	Rc	Rd	Ra	Jump to Address in STKO	Stack Ref
N+1	—	Ra+1	K	Ra	Rb	Rc	Rd	—		(Loop)
N	1 1 0 0	J	K	Ra	Rb	Rc	Rd	D	Pop Stack;	End
N+1	—	D+1	K	Rb	Rc	Rd	Ra	—	Jump to Address on D	Loop
N	1 1 0 1	J	K	Ra	Rb	Rc	Rd	D	Jump to Address on D;	JSR D
N+1	—	D+1	K	J	Ra	Rb	Rc	—	Push μPC	
N	1 1 1 X	J	K	Ra	Rb	Rc	Rd	D	Jump to Address on D	JMP D
N+1	—	D+1	K	Ra	Rb	Rc	Rd	—		

X = Don't care, 0 = LOW, 1 = HIGH, Assume C_n = HIGH

Note: STK0 is the location addressed by the stack pointer.

Figure 6. Output and Internal Next-Cycle Register States for 2909/2911.

Operation of the 2909/2911

Figure 5 lists the select codes for the multiplexer. The two bits applied from the microword register (and additional combinational logic for branching) determine which data source contains the address for the next microinstruction. The contents of the selected source will appear on the Y outputs. Figure 5 also shows the truth table for the output control and for the control of the push/pop stack. Figure 6 shows in detail the effect of S_0, S_1, $\overline{FE}$ and PUP on the 2909. These four signals define what address appears on the Y outputs and what the state of all the internal registers will be following the clock LOW-to-HIGH edge. In this illustration, the microprogram counter is assumed to contain initially some word J, the address register some word K, and the four words in the push/pop stack contain R_a through R_d.

Figure 7 illustrates the execution of a subroutine using the 2909. The configuration of Figure 3 is assumed. The instruction being executed at any given time is the one contained in the microword register (μWR). The contents of the μWR also controls (indirectly, perhaps) the four signals S_0, S_1, FE, and PUP. The starting address of the subroutine is applied to the D inputs of the 2909 at the appropriate time.

In the columns on the left is the sequence of microinstructions to be executed. At address J+2, the sequence control portion of the microinstruction contains the command "Jump to subroutine at A". At the time T_2, this instruction is in the μWR, and the 2909 inputs are set-up to execute the jump and save the return address. The subroutine address A is applied to the D inputs from the μWR and appears on the Y outputs. The first instruction of the subroutine, I(A), is accessed and is at the inputs of the μWR. On the next clock transition, I(A) is loaded into the μWR for execution, and the return address J+3 is pushed onto the stack. The return instruction is executed at T_5. Figure 8 is a similar timing chart showing one subroutine linking to a second, the latter consisting of only one microinstruction.

Control Memory

EXECUTE CYCLE	MICROPROGRAM ADDRESS	MICROPROGRAM SEQUENCER INSTRUCTION	EXECUTE CYCLE CLOCK / SIGNALS		T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
	J−1	—	2909	S_1,S_0	0	0	3	0	0	3	2	0	2	0
T_0	J	—	Inputs	$\overline{FE}$	H	H	L	H	H	L	L	H	L	H
T_1	J+1	—	(from	PUP	X	X	H	X	X	H	L	X	L	X
T_2	J+2	JSR A	μWR)	D	X	X	A	X	X	B	X	X	X	X
T_9	J+3	—												
		—		μPC	J+1	J+2	J+3	A+1	A+2	A+3	B+1	A+4	A+5	J+4
	—	—		STK0	—	—	—	J+3	J+3	J+3	A+3	J+3	J+3	
	—	—	Internal	STK1	—	—	—	—	—	—	J+3	—		
	—	—	Registers	STK2	—	—	—	—	—	—	—	—		
	—	—		STK3	—	—	—	—	—	—	—	—		
	—	—												
T_3	A	I(A)	2909 Output	Y	J+1	J+2	A	A+1	A+2	B	A+3	A+4	J+3	J+4
T_4	A+1	—												
T_5	A+2	JSR B	ROM Output	(Y)	I(J+1)	JSR A	I(A)	I(A+1)	JSR B	RTS	I(A+3)	RTS	I(J+3)	I(J+4)
T_7	A+3	—												
T_8	A+4	RTS												
	—	—	Contents of μWR (Instruction being executed)	μWR	I(J)	I(J+1)	JSR A	I(A)	I(A+1)	JSR B	RTS	I(A+3)	RTS	I(J+3)
	—	—												
	—	—												
	—	—												
T_6	B	RTS												

Figure 7. Subroutine Execution.

Control Memory

EXECUTE CYCLE	MICROPROGRAM ADDRESS	MICROPROGRAM SEQUENCER INSTRUCTION	EXECUTE CYCLE CLOCK / SIGNALS		T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
	J−1	—	2909 Inputs (from μWR)	S_1,S_0	0	0	3	0	0	2	0	0		
T_0	J	—		$\overline{FE}$	H	H	L	H	H	L	H	H		
T_1	J+1	—		PUP	X	X	H	X	X	L	X	X		
T_2	J+2	JSR A		D	X	X	A	X	X	X	X	X		
T_6	J+3	—	Internal Registers	μPC	J+1	J+2	J+3	A+1	A+2	A+3	J+4	J+5	—	—
T_7	J+4	—		STK0	—	—	—	J+3	J+3	J+3	—	—	—	—
	—	—		STK1	—	—	—	—	—	—	—	—	—	—
	—	—		STK2	—	—	—	—	—	—	—	—	—	—
	—	—		STK3	—	—	—	—	—	—	—	—	—	—
	—	—												
T_3	A	1(−A)	2909 Output	Y	J+1	J+2	A	A+1	A+2	J+3	J+4	J+5		
T_4	A+1	—	ROM Output	(Y)	I(J+1)	JSR A	I(A)	I(A+1)	RTS	I(J+3)	I(J+4)	I(J+5)		
T_5	A+2	RTS												
	—	—												
	—	—												
	—	—	Contents of μWR (Instruction being executed)											
	—	—		μWR	I(J)	I(J+1)	JSR A	I(A)	I(A+1)	RTS	I(J+3)	I(J+4)		
	—	—												
	—	—												

Figure 8. Two Nested Subroutines. Routine B is Only One Instruction.

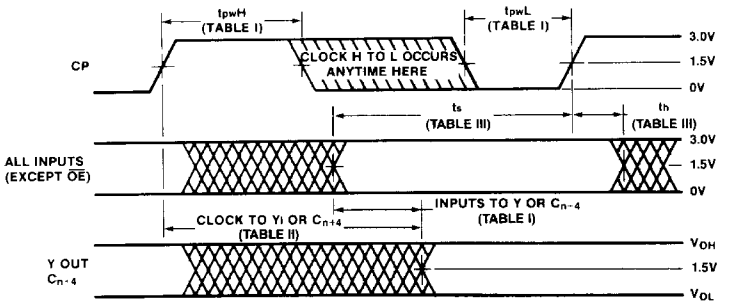


Figure 12. Switching Waveforms. See Tables for Specific Values.