

4-BIT EXPANDABLE BIPOLAR MICROCONTROLLER

5701/6701

**Monolithic
Memories**
INCORPORATED

5701
6701

MILITARY	COMMERCIAL
X	
	X

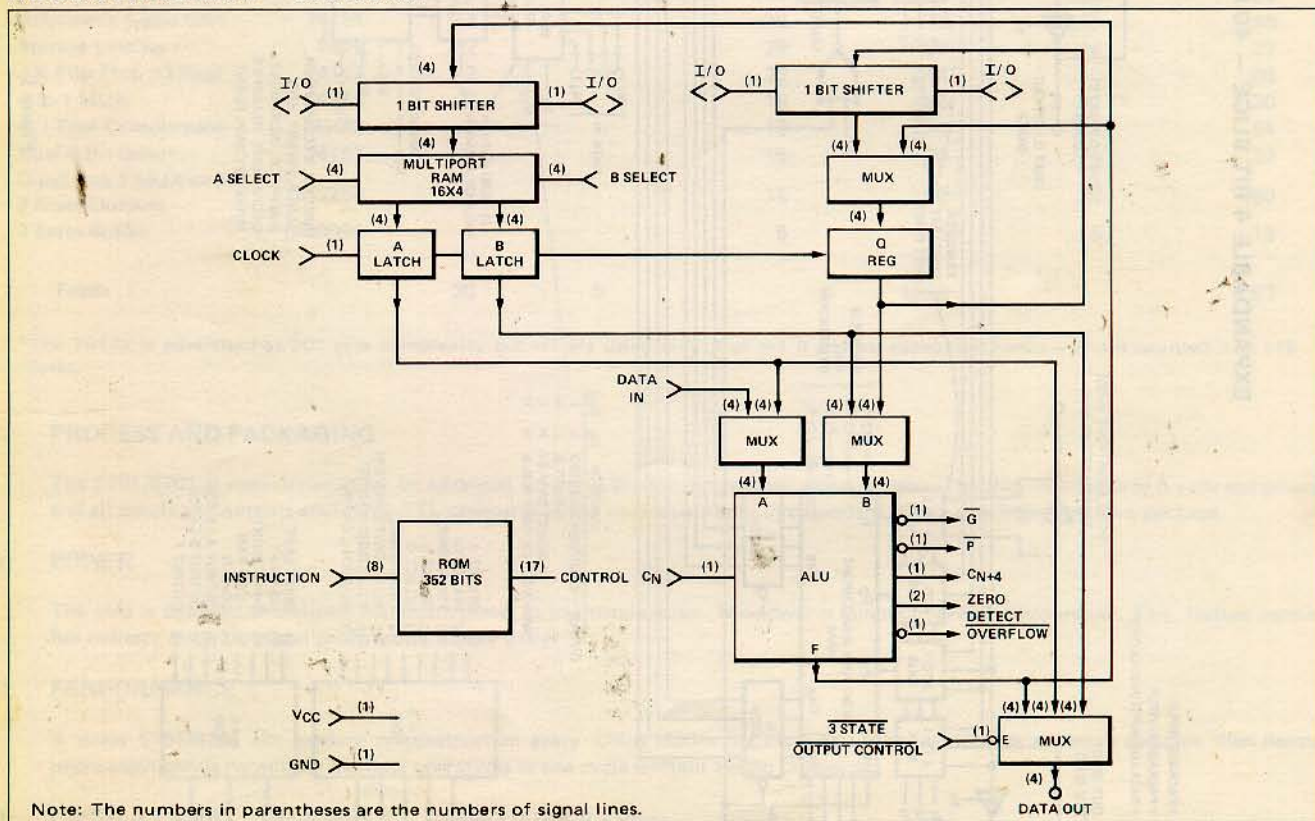
PRODUCT FEATURES

- Complete 4-Bit Bipolar LSI Processor Slice on a Single Chip.
- Replaces 25 TTL MSI Packages and Saves 5.5 Watts.
- 1000 Gate Complexity Schottky LSI – Single Layer Metal.
- 36 Instructions – Arithmetic, Logic, and Shifting Capability with Overflow Detection. Active High or Active Low Logic.
- 16 Directly Addressable, Two-Port, General Purpose Accumulators – Full 2 Address Capability, Some 3 Register Operations
- A Separate Q-Register Useful as a Scratchpad or Accumulator Extension. Direct Data in and Accumulator Operations.
- Separate Low Fan in Input Bus and 3 State Output Bus.
- Expandable to Handle N Bit Words with Full Carry Look-Ahead.
- 175 ns Cycle (6701) Which Can Perform Multiple Nano-instructions such as Subtract, Shift, and Store in One Cycle.

APPLICATIONS

- The Ideal Product for Upgrading or Replacing Existing Central Processing Units and Maintaining the Existing Software. The 5701/6701 can be Microprogrammed to Efficiently Emulate (Simulate) Most Machines.
- Hard Wired Controllers – Tape and Disk Controllers – Data Concentrators.
- Point of Sale Terminals, Special Purpose Processors.
- Process and Machine Control.
- Word Processing and Navigation Systems.
- Intelligent Terminals and Game Machines.
- Traffic Control and Communications Systems.
- Upgrade Systems Using the 74181, 9340, 9341, 74S281 Arithmetic Logic Units

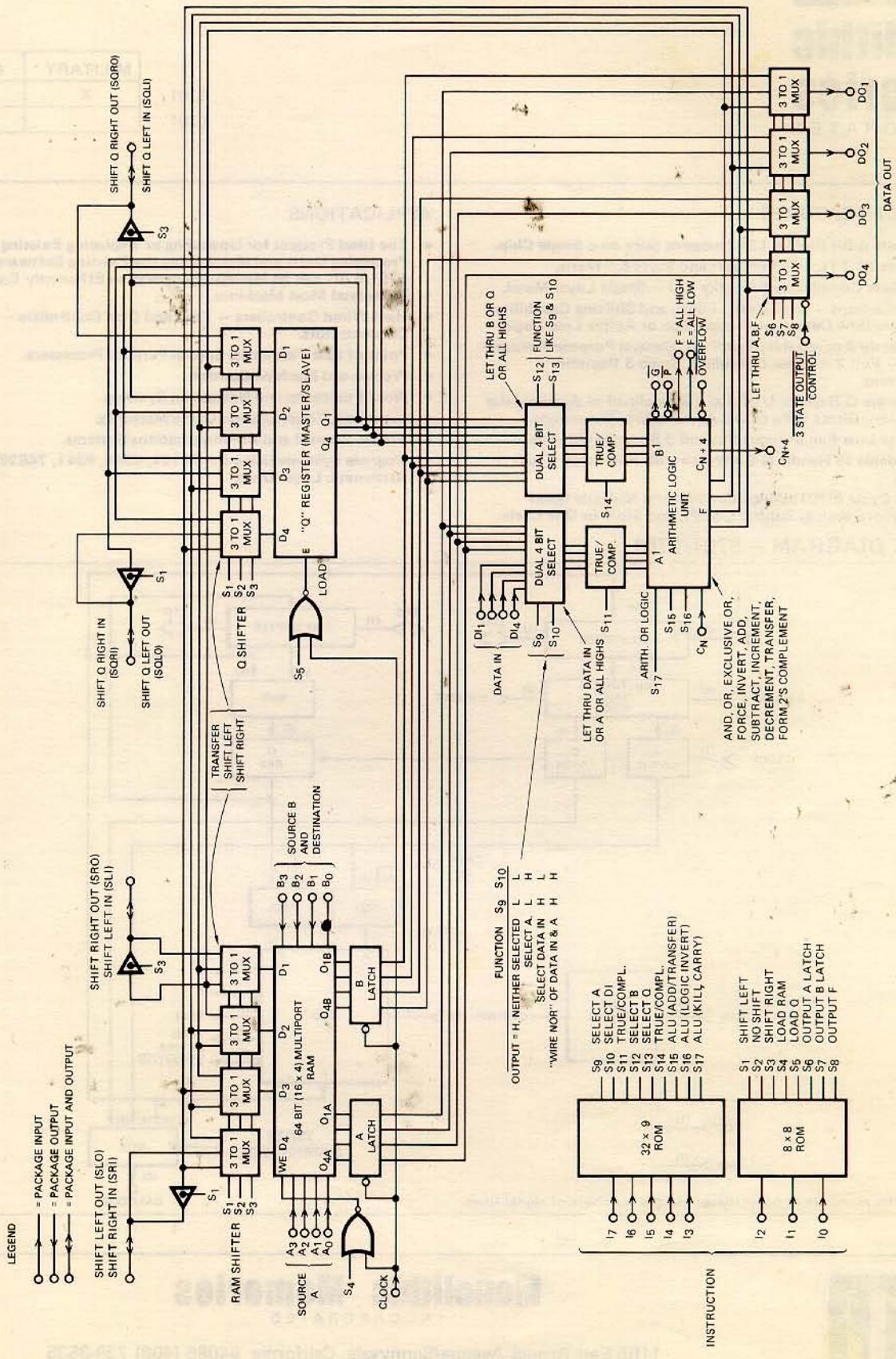
BLOCK DIAGRAM – 5701/6701



Monolithic Memories
INCORPORATED

1165 East Arques Avenue/Sunnyvale, California 94086 (408) 739-3535
TWX 910-339-9229

EXPANDABLE 4 BIT SLICE — 40 PIN PKG.



LEGEND

- = PACKAGE INPUT
- = PACKAGE OUTPUT
- = PACKAGE INPUT AND OUTPUT

FUNCTION S₉ S₁₀

- OUTPUT = H, NEITHER SELECTED L L
- SELECT A L L H
- SELECT DATA IN H L L
- "WIRE NOR" OF DATA IN & A H H

- S₉ SELECT A
- S₁₀ SELECT DI
- S₁₁ TRUE/COMPL.
- S₁₂ SELECT B
- S₁₃ SELECT Q
- S₁₄ TRUE/COMPL.
- S₁₅ ALU (ADD/TRANSFER)
- S₁₆ ALU (LOGIC INVERT)
- S₁₇ ALU (KILL, CARRY)

- S₁ SHIFT LEFT
- S₂ NO SHIFT
- S₃ SHIFT RIGHT
- S₄ LOAD RAM
- S₅ LOAD Q
- S₆ OUTPUT A LATCH
- S₇ OUTPUT B LATCH
- S₈ OUTPUT F

Figure 1

I. INTRODUCTION

A) GENERAL DESCRIPTION

The 5701/6701 is designed to be used as a 4-bit processor slice of a conventional central processing unit (C.P.U.). It can also be used in peripheral controllers, (tape, disk, etc.,) or as the heart of a microprocessor, terminal, or computer. It is a single chip bipolar LSI device which replaces more than 1000 TTL gates.

The two address capability (ability to work on two accumulators at once) and the powerful nano-instructions permit design of sub 1 microsecond cycle time hard wired C.P.U.'s or efficient emulation (imitation) of conventional machines using off chip ROMs for microprogramming.

The 5701/6701 will handle the data flow section of most computers since it is expandable to handle any word length in increments of 4 bits without significant speed degradation (look-ahead outputs are available). The 16 on-chip general purpose accumulators implement the type of C.P.U. usually found only in high performance top of the line 16-bit minicomputers or 24 or 32-bit computers. It can be thought of as a general purpose 4-bit register and arithmetic logic unit with a separate A operand, B operand, data-in, and data-out ports. Additional accumulators or registers if required can be added with off chip packages tied to the data in pins.

B) TTL EQUIVALENT

The 5701/6701 is similar in function to the 25 TTL MSI packages listed below. It saves 375 I/O pins, 5.6 watts and 30 square inches of board area.

TABLE 1

Function	TTL #	#14 Pin or #16 Pin Pkgs.	#24 Pin Pkgs.	Advertised Gate Complexity (Each Pkg.)	Gate Complexity Total	Typical Power Each (Watts)	Total Power (Watts)
32 x 9 & 8 x 8 ROMs	7488	3		70	210	.50	1.50
16 x 4 Multiport RAM	74172		4	110*	440	.56	2.24
Arithmetic Logic Unit	74181		1	75	75	.55	.55
Storage Latches	7475	2		28	56	.16	.32
J-K Flip Flop (Q Reg)	74107	2		22	44	.10	.20
4 to 1 MUX	74153	6		16	96	.20	1.20
O/I True Complement	74H87	2		18	36	.27	.54
Dual 4 Bit Select	74157	2		15	30	.15	.30
Quad 2 to 1 MUX with 3 State Outputs	74S257	2		15	30	.30	.60
3 State Buffer	DM8094	1		5	5	.18	.18
Totals		20	5		1022		6.63

*The 74172 is advertised at 201 gate complexity but we are using only 2 of the 3 address capability, hence we have counted it as 110 gates.

C) PROCESS AND PACKAGING

The 5701/6701 is manufactured by an advanced Schottky bipolar single layer metal process. The chip requires only 5 volts and ground and all inputs and outputs are totally TTL compatible. The chip is packaged in a standard 40-pin dual in-line ceramic package.

D) POWER

The chip is designed to dissipate maximum power at low temperature. The power is minimum at high temperature. This feature permits full military range parts and easier power supply design.

E) PERFORMANCE

A single 5701/6701 can execute one instruction every 175ns (230ns for the 5701). The instructions are more complex than normal micro-instructions permitting multiple operations in one cycle without timing problems.

II. OPERATION — See Figure 1 — Expandable 4-Bit Slice

A) BLOCK DIAGRAM

A detailed block diagram of the chip is shown in Figure 1. Note the legend used in the upper left hand corner for package inputs and outputs. The logic and control ones are shown as they are implemented on the chip even though fewer control inputs might be required by discrete logic devices. The dual 4-bit selects, at the input to the arithmetic logic unit, for example, have two "S" input control lines rather than the 74157 TTL equivalent which has one control line for letting through the left four bits or the right four bits on the output 4 bits.

B) ROM (LOWER LEFT CORNER)

Two on chip ROMs (352 bits total) are used to translate the eight (8) instruction lines I_0 to I_7 into 17 on chip control lines (labeled S_1 through S_{17}) which open and close data paths required to execute an instruction.

C) MULTI PORT RAM (UPPER LEFT CORNER)

A 16 word by 4 bit multiple port memory is used to fetch two operands at the same time. The RAM is double decoded (4 A address pins and 4B address pins and has one set of 64 storage cells). We could, for example, read from 0101 on the A address (file #5) and read from 0001 on the B address (file #1) at the same time. The B side of the RAM can be read out or written into independent of the address on the A side. The A side can only be read. The RAM must be loaded via the B side. The RAM, if it is enabled, is loaded when the clock is low. The duration of time the clock is low, is the write enable pulse width required to switch the RAM. If the clock is held high, operations can be performed and the results examined, but no results are stored.

D) LATCHES (CENTER LEFT)

The latches on the RAM outputs are the zero delay type (like 7475) which let the data into the latches appear on the latch outputs until the clock goes low and then hold the data. The latches permit parallel accessing of the RAM and ROM without two delays (one for the ROM and one for the RAM), since the access time of the ROM is masked by the delay through the RAM. The latches eliminate race conditions when the RAM data is fetched and updated in one cycle.

E) ARITHMETIC LOGIC UNIT (LOWER CENTER)

Input multiplexers into the arithmetic logic unit (A.L.U.) under ROM control permit the entry of data in or the A channel of the RAM into the A port of the ALU, and the B channel of the RAM or the Q register into the B port of the ALU.

The ALU is of the conventional type except 0/1 true complement elements have been put in the input ports permitting the realization of a totally symmetrical ALU (i.e., A minus B or B minus A). Overflow detection and two zero detect pin (one for positive and one for negative logic) are also included.

F) OUTPUT MULTIPLEXERS (LOWER RIGHT HAND CORNER)

The 3 to 1 output multiplexers under ROM control let through the ALU output, the A latch output, the B latch output on the data out pins. The output multiplexers are three-state outputs controlled by the three-state output control pin. The three-state outputs permit processing to be performed in the 5701/6701 without tying up the data out bus and allow one bus system to tie 5701/6701 data in pins to data out pins. The outputs are enabled when the three-state enable pin is low.

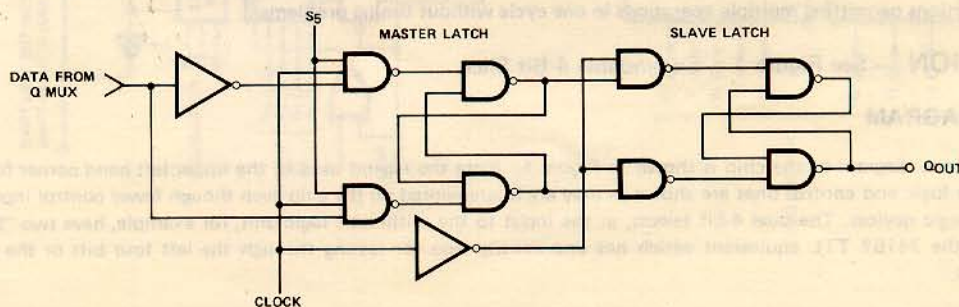
G) SHIFT MULTIPLEXERS (UPPER LEFT AND UPPER RIGHT)

The four 3 to 1 mux's on top of the Q register permit the ALU output bus F to be loaded into the Q register. The Q register can also be shifted on itself one place left or right. The four 3 to 1 multiplexers above the RAM permit a transfer or 1 bit left or right shift of the F bus before storage in the RAM. Both the RAM shifter and the Q shifter employ bi-directional shift in/shift out pins to permit expansion to more than 4 bits.

H) Q REGISTER (UPPER RIGHT)

The Q register can function as an accumulator extension register. It would normally be used to hold the least significant half of the double length product of a multiplication or as a storage register to catch the bits shifted off the beginning or end of a word during left or right shifting. In this mode of operation, the shift out pin of the least significant bit of the RAM shifter would be tied to the shift in pin of the most significant bit of the Q register. The Q register can be shifted on itself or be loaded from the ALU bus. The Q shift control pins are in common with RAM shift controls permitting simultaneous RAM and Q shifting. It can also be used as a program counter or scratchpad.

The Q register is a master slave flip-flop. Data is loaded into the master while the clock is low (assuming it is enabled by the ROM) and transferred from the master to the slave when the clock goes high. The Q register is not an edge-triggered latch. Whatever data is present before the clock goes high (assuming the setup time is obeyed) will be loaded into Q. If the clock is held high operations can be performed but no results are stored.



III. PACKAGE PINS

Figure 2 shows a listing of the 40 pin I/O. The 5701/6701 uses a standard 40 lead ceramic dip approximately 2.00 inches long and 0.60 inch wide. See Figure 3 for the package details. NOTE THAT VCC AND GROUND ARE NOT ON THE CONVENTIONAL END PINS. This was necessary because the package current would produce too high a voltage drop in the package lead resistance out to the end pins to meet the V_{OL} requirements.

PIN CONFIGURATION

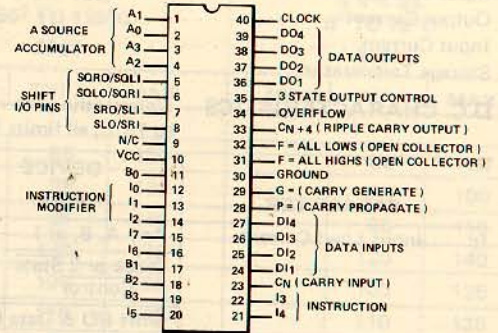


Figure 2.

PACKAGE OUTLINE

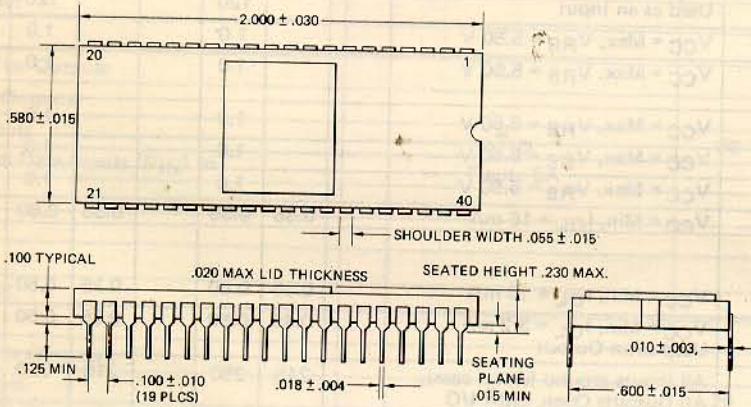


Figure 3.

40 Pin Ceramic (Side Braze)

Θ_{JA} (thermal resistance from junction to ambient soldered to a printed circuit board in still air) $\approx 35^\circ\text{C/watt}$

Θ_{JC} (thermal resistance from junction to case with freon as a heat sink) $\approx 20^\circ\text{C/watt}$

The circuit in Figure 4 below puts the device in an instruction where the state of all outputs is known. Any other set of levels on the inputs may result in undefined output states and possible damage to the device under burn-in conditions. The other instructions involve the use of internal register whose initial conditions, after power-up, are undefined. The burn-in circuit shown comes closest to the usual reverse burn-in circuits of logic devices. The unit is in the force LLLL mode with the shift I/O pins disabled, the data outputs disabled, and with a single accumulator always selected.

BURN-IN CIRCUIT

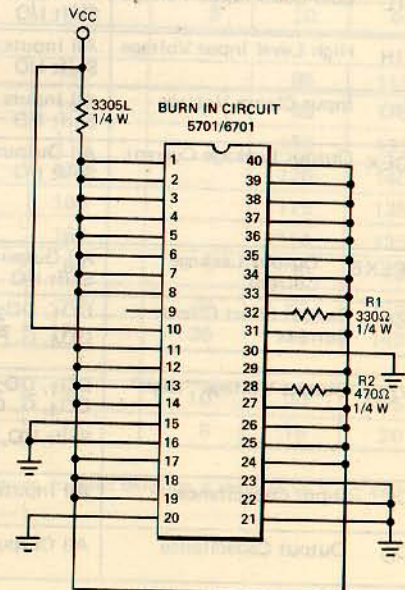


Figure 4.

ELECTRICAL PARAMETERS

ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V_{CC}	-0.5 to 7.0 V
Input Voltage	-1.0 to 5.5 V
Output Current	100 mA
Input Current	-20 to 5 mA
Storage Temperature	-65 to +150 °C

Stresses above or extended time at Absolute Maximum Ratings may cause permanent damage or affect device reliability.

D.C. CHARACTERISTICS – Unless otherwise indicated, all limits for the 6701 are guaranteed for $5\text{ V} \pm 5\%$ in a free air temperature of 0° to 75°C ; all limits for the 5701 are guaranteed for $5\text{ V} \pm 10\%$ in a free air temperature of -55° to 125°C .

PARAMETER	DEVICE PINS	CONDITIONS	5701			6701			UNITS
			MIN.	TYP. ¹	MAX.	MIN.	TYP. ¹	MAX.	
I_F Input Load Current	Any A, B, or I	$V_{CC} = \text{Max}, V_F = .45\text{ V}$			-250			-250	μA
	Clock or 3 State Control	$V_{CC} = \text{Max}, V_F = .45\text{ V}$			-250			-250	μA
	Shift I/O & Data In	$V_{CC} = \text{Max}, V_F = .45\text{ V}$			-0.80			-0.80	mA
	C_N	$V_{CC} = \text{Max}, V_F = .45\text{ V}$			-4.80			-4.80	mA
I_R Input Leakage Current	Any A, B, or I	$V_{CC} = \text{Max}, V_R = 2.40\text{ V}$			40			25	μA
	Clock or 3 State Control	$V_{CC} = \text{Max}, V_R = 2.40\text{ V}$			40			25	μA
	Shift I/O	Used as an Input			100			100	μA
	Data Inputs	Used as an Input			20			20	μA
	C_N	Used as an Input			120			120	μA
I_{RB} Input Leakage Current	Any A, B, or I	$V_{CC} = \text{Max}, V_{RB} = 5.50\text{ V}$			1.0			1.0	mA
	Clock or 3 State Control	$V_{CC} = \text{Max}, V_{RB} = 5.50\text{ V}$			1.0			1.0	mA
	Shift I/O	$V_{CC} = \text{Max}, V_{RB} = 5.50\text{ V}$			1.0			1.0	mA
	Data Input	$V_{CC} = \text{Max}, V_{RB} = 5.50\text{ V}$			1.0			1.0	mA
	C_N	$V_{CC} = \text{Max}, V_{RB} = 5.50\text{ V}$			1.0			1.0	mA
V_{OL} Low Level Output Voltage	Data Outputs, \bar{G} , C_{N+4} , F = All High, F = All Low	$V_{CC} = \text{Min}, I_{OL} = 16\text{ mA}$		0.35	0.50		0.35	0.50	V
	P, Overflow	$V_{CC} = \text{Min}, I_{OL} = 10\text{ mA}$		0.35	0.50		0.35	0.50	V
	Shift I/O	$V_{CC} = \text{Min}, I_{OL} = 3.2\text{ mA}$ Used as an Output		0.35	0.50		0.35	0.50	V
I_{CC} Power Supply Current	V_{CC}, Grd	All inputs ground (worst case), All Outputs Open, Shift I/O Open Temp = Max		215	250		215	250	mA
		All inputs ground (worst case), All Outputs Open, Shift I/O Open Temp = Min		230	280		230	280	mA
V_{IL} Low Level Input Voltage	All Inputs and Shift I/O	$V_{CC} = 5.00\text{ V}$			0.80			0.80	V
V_{IH} High Level Input Voltage	All Inputs and Shift I/O	$V_{CC} = 5.00\text{ V}$	2.0			2.0			V
V_{IC} Input Clamp Voltage	All Inputs and Shift I/O	$V_{CC} = \text{Min}, I_{IC} = -5.0\text{ mA}$		-1.0	-1.5		-1.0	-1.5	V
I_{CEX} Output Leakage Current	All Outputs and Shift I/O	$V_{CC} = \text{Max}, V_{CEX} = 2.40\text{ V}$, High Stored or Disabled			100			100	μA
		$V_{CC} = \text{Max}, V_{CEX} = 0.45\text{ V}$ Disabled			-100			-100	μA
I_{CEXB} Output Leakage Current	All Outputs and Shift I/O	$V_{CC} = \text{Max} = V_{CEXB}$ High Stored or Disabled			1			1	mA
I_{SC} Output Short Circuit Current	$DO_1, DO_2, DO_3, DO_4, \bar{G}, \bar{P}, \bar{OVFL}$	$V_{OUT} = 0\text{ V}, V_{CC} = 5\text{ V}$ Only one Output at a time should be Shorted	5	15	90	5	15	90	mA
V_{OH} Output Voltage "High"	$DO_1, DO_2, DO_3, DO_4, \bar{G}, C_{N+4}$	$I_O = -3.2\text{ mA}, V_{CC} = \text{Min}$ High Stored	2.4	3.5		2.4	3.5		V
	Shift I/O, \bar{P}, \bar{OVFL}	$I_O = -500\ \mu\text{A}, V_{CC} = \text{Min}$ High Stored	2.4	3.5		2.4	3.5		V
C_I Input Capacitance	All Inputs	$V_{CC} = 5.0\text{ V}, V_I = 2.0\text{ V}, 25^\circ\text{C}$, 1 MHz		8			8		pF
C_O Output Capacitance	All Outputs	$V_{CC} = 5.0\text{ V}, V_O = 2.0\text{ V}, 25^\circ\text{C}$, 1 MHz		8			8		pF

1. Typical values are measured at 5.0 V and 25°C .

A.C. CHARACTERISTICS*

PARAMETER	SYMBOL/CONDITIONS	TIME IN NANoseconds					
		5 V ± 10% -55° TO 125°C			5 V ± 5% 0° TO 75°C		
		5701			6701		
		MIN.	TYP. ¹	MAX.	MIN.	TYP. ¹	MAX.
Delay from RAM Address (A _N or B _N) to	Figure 5A Clock High						
\overline{G}			85	130		85	100
\overline{P}			85	130		85	100
C _{N+4}			95	140		95	110
Zero Detect			120	170		120	140
Overflow			100	155		100	125
RAM Shifter Outputs			110	165		110	135
Data Outputs (bypass ALU)	T _{C1}	20	85	130	20	85	100
Data Outputs (through ALU)	T _{AA}	40	110	170	40	110	140
Delay from Instruction Input (I _N) to	Figure 5A						
\overline{G}			55	95		55	75
\overline{P}			55	95		55	75
C _{N+4}			60	100		60	80
Zero Detect			80	120		80	105
Overflow			75	100		75	95
RAM Shifter Outputs			75	125		75	95
Q Shifter Outputs			25	60		25	40
Data Outputs	T _{ID}	30	85	120	15	85	100
Delay from Data Inputs (D _{IN}) to	Figure 5A						
\overline{G}			25	60		25	40
\overline{P}			25	60		25	40
C _{N+4}			25	65		25	45
Zero Detect			20	95		20	75
Overflow			40	75		40	55
RAM Shifter Outputs			50	90		50	70
Data Outputs	T _{DD}	20	50	95	10	50	75
Delay from Carry In (C _N) to	Figure 5A						
C _{N+4}			15	30		15	25
Zero Detect			30	45		30	40
Overflow			25	40		25	35
RAM Shifter Outputs			25	40		25	35
Data Outputs		5	30	45	5	30	40
Delay from Clock (low to high) to	A _N , B _N , I _N , D _{IN} constant						
\overline{G}				145		95	115
\overline{P}				145		95	115
C _{N+4}				115		105	125
Zero Detect				170		120	140
Overflow				165		115	135
RAM Shifter Outputs				165		110	135
Q Shifter Outputs				60		30	40
Data Outputs (bypass ALU)	T _{C2}	20	60	145	20	95	115
Data Outputs (through ALU)				170	30	115	140
Delay from 3 State Control to	All Inputs constant						
Low Data Outputs	T _{EA}	10	20	30	10	20	25
High Impedance Outputs	T _{ER}	5	14	25	5	15	20

*The maximum of all possible input and output conditions is specified with outputs sinking maximum current through a resistor to V_{CC} and driving a 30 pF load (15 pF on Shift I/O) to ground.

¹ Typical values are measured at 5.0 V and 25°C.

A.C. CHARACTERISTICS* - Cont'd.

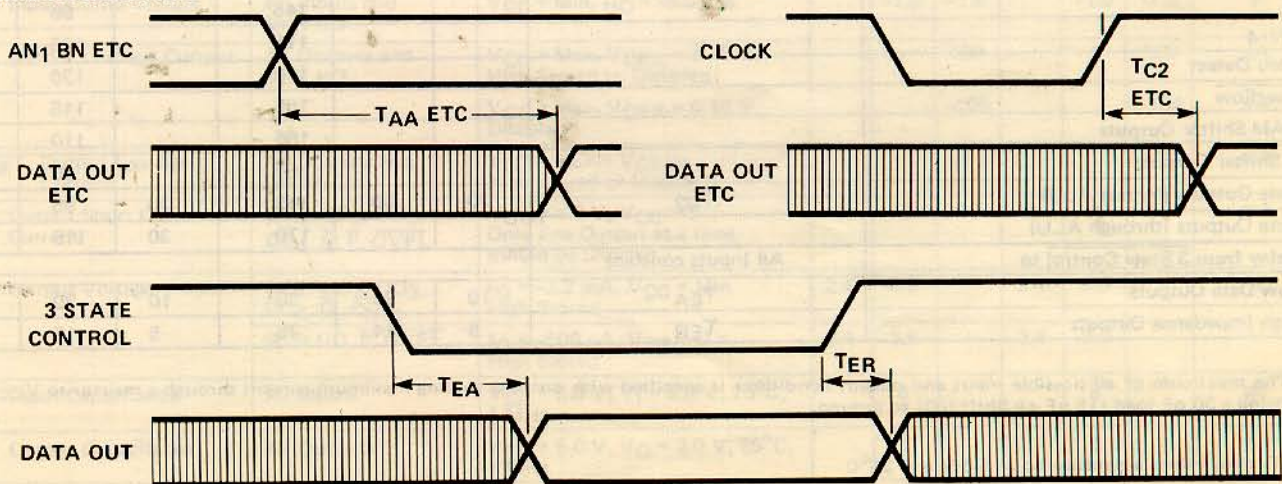
PARAMETER	SYMBOL/CONDITIONS	TIME IN NANOSECONDS					
		5 V ± 10% -55° TO 125°C			5 V ± 5% 0° TO 75°C		
		MIN.	TYP. ¹	MAX.	MIN.	TYP. ¹	MAX.
Setup Time Before the High to Low Transition of Clock	Figure 5B						
A _N	T _{AS}	115	60		85	60	
B _N	T _{BS1}	120	60		90	60	
I ₀ , I ₁ , I ₂	T _{IS1}	30	5		15	5	
Setup Time Before the Low to High Transition of Clock	Figure 5B						
A _N , B _N	T _{BS2}	240	130		185	130	
C _N	T _{CNS}	95	45		70	45	
I ₃ , I ₄ , I ₅ , I ₆ , I ₇	T _{IS2}	150	90		120	90	
Data Inputs (D _{IN})	T _{DS}	95	50		70	50	
RAM Shifter Inputs	T _{RS}	60	30		45	30	
Q Shifter Inputs	T _{QS}	45	15		30	15	
Hold Time After the Low to High Transition of Clock	Figure 5B						
A _N	T _{AH}	-T _{CW} -10	-T _{CW} -25		-T _{CW} -15	-T _{CW} -10	
B _N	T _{BH}	-10	-20		-10	-20	
I ₀ , I ₁ , I ₂	T _{IH1}	-20	-40		-20	-40	
I ₃ , I ₄ , I ₅ , I ₆ , I ₇	T _{IH2}	-30	-60		-30	-60	
C _N	T _{CNH}	-20	-40		-20	-40	
Data Inputs (D _{IN})	T _{DH}	-25	-50		-25	-50	
RAM Shifter Inputs	T _{RH}	-25	-50		-25	-50	
Q Shifter Inputs	T _{QH}	-15	-30		-15	-30	
System Parameters	Figure 5B						
Clock Pulse Width (Low Time)	T _{CW}	80	25		50	25	
Minimum Microinstruction Cycle Time	T _{CYCLE} = T _{BS2} + T _{BH}	230	110	∞	175	110	∞

*The maximum of all possible input and output conditions is specified with outputs sinking maximum current through a resistor to V_{CC} and driving a 30 pF load (15 pF on Shift I/O) to ground.

¹ Typical values are measured at V_{CC} = 5.0 V and 25°C.

TIMING DIAGRAMS

Figure 5A - Delays



LEGEND: Shaded areas indicate don't care conditions or permitted timing tolerances.

TIMING DIAGRAMS - Cont'd.

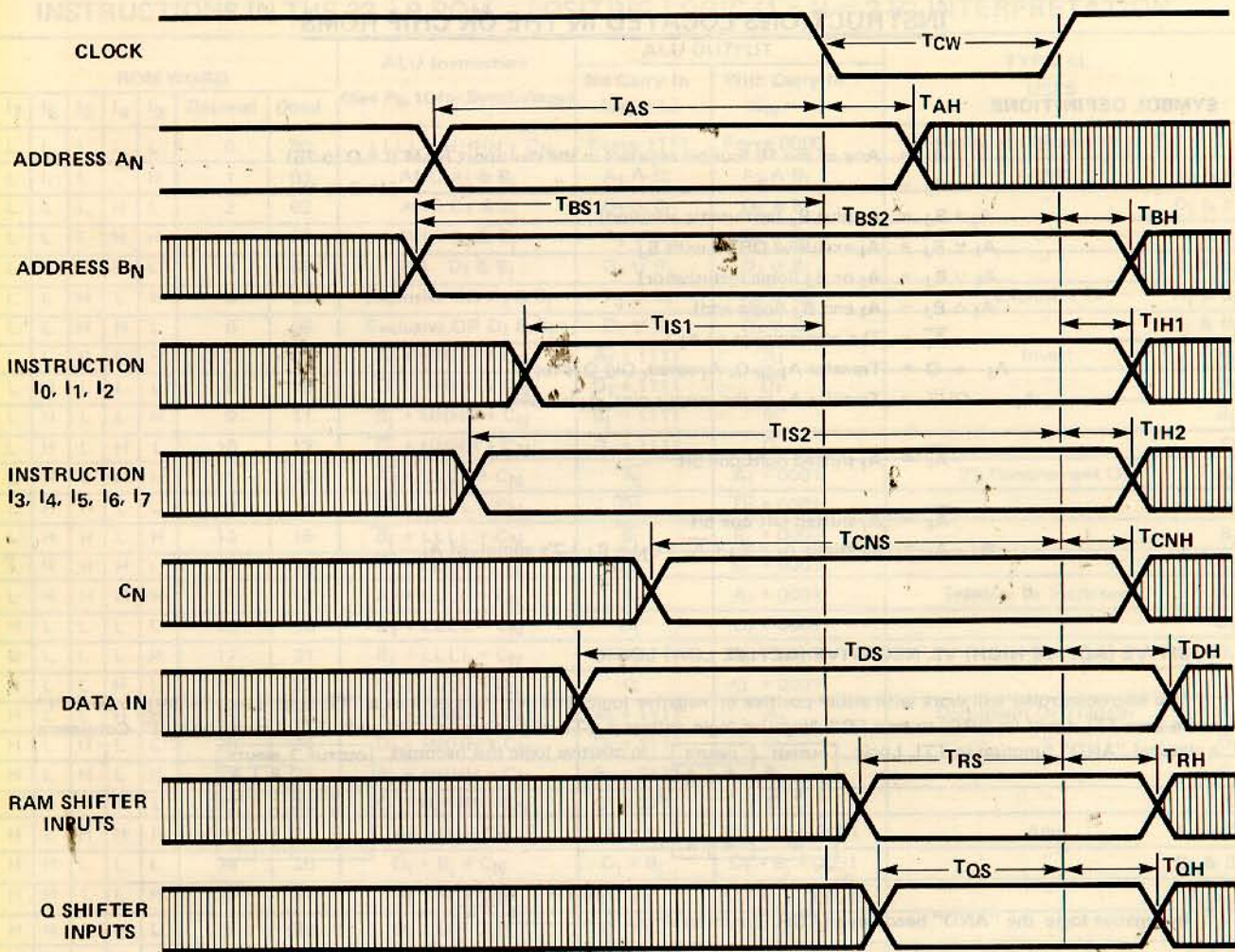


Figure 5B - Set-up and Hold Times

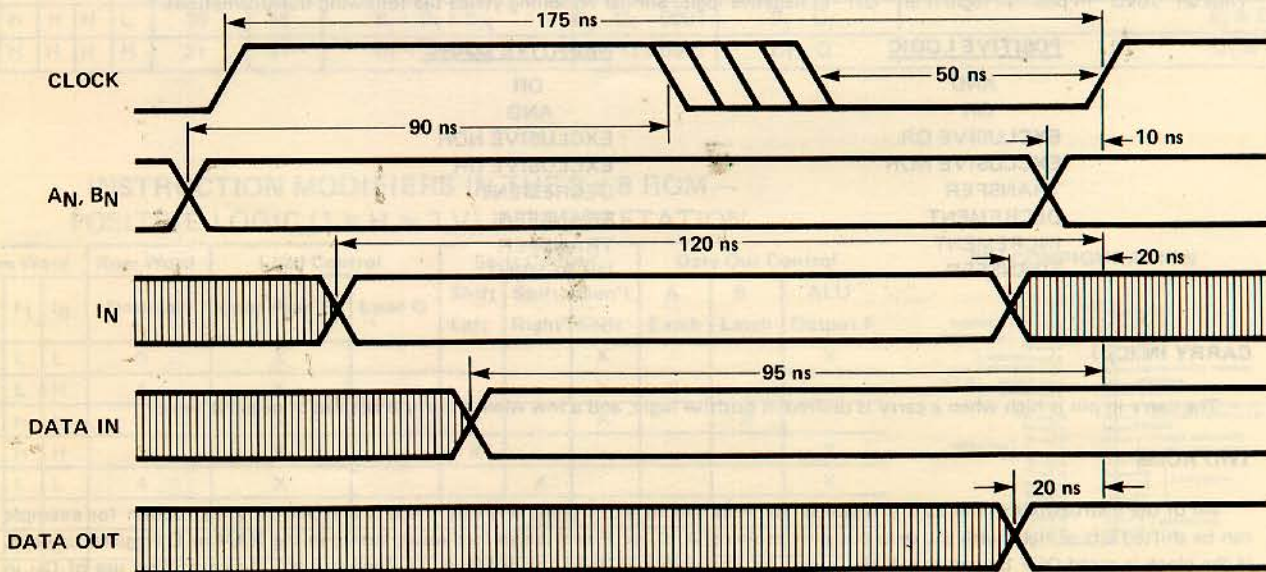


Figure 5C - Sample Minimum Cycle Time for 6701

INSTRUCTIONS LOCATED IN THE ON CHIP ROMS

SYMBOL DEFINITIONS

- A_I = Any of the 16 four bit registers in the multiport RAM ($I = 0$ to 15)
 B_J = " " " " ($J = 0$ to 15)
 $A_I + B_J$ = A_I plus B_J (arithmetic addition)
 $A_I \nabla B_J$ = A_I exclusive OR'ed with B_J
 $A_I \vee B_J$ = A_I or B_J (logic inclusive or)
 $A_I \wedge B_J$ = A_I and B_J (logic and)
 $\overline{A_I}$ = The complement of A_I
 $A_I \rightarrow Q$ = Transfer A_I to Q, A_I saved, Old Q is lost
 $A_I \rightarrow \text{OUT}$ = Transfer A_I to the output pins, A_I is saved
 $\xrightarrow{A_I}$ = A_I shifted right one bit
 $\xleftarrow{A_I}$ = A_I shifted left one bit
 $B_J - A_I$ = B_J minus $A_I = B_J + \overline{A_I} + C_N = B_J + 2$'s compl. of A_I

POSITIVE (ACTIVE HIGH) VS. NEGATIVE (ACTIVE LOW) LOGIC

The Microcontroller will work with either positive or negative logic. Positive logic defines a TTL High Level ($\approx 3V$) to be a "1" and a Low TTL Level ($\approx \text{GRD}$) to be a "0". Negative logic defines a TTL High to be a "0" and a TTL Low to be a "1". Consider a classical "AND" function in TTL Logic

OUTPUT	INPUTS	
	B	A
L	L	L
L	L	H
L	H	L
H	H	H

in positive logic this becomes

OUTPUT	INPUTS	
	B	A
0	0	0
0	0	1
0	1	0
1	1	1

In negative logic the "AND" becomes an "OR" function since:

OUTPUT	INPUTS	
	B	A
1	1	1
1	1	0
1	0	1
0	0	0

Thus an "AND" in positive logic is an "OR" in negative logic. Similar reasoning yields the following transformations.

POSITIVE LOGIC

AND
 OR
 EXCLUSIVE OR
 EXCLUSIVE NOR
 TRANSFER
 DECREMENT
 INCREMENT
 TRANSFER

NEGATIVE LOGIC

OR
 AND
 EXCLUSIVE NOR
 EXCLUSIVE OR
 DECREMENT
 TRANSFER
 TRANSFER
 INCREMENT

CARRY IN (C_N)

The carry in pin is high when a carry is desired in positive logic, and a low when a carry is desired in negative logic.

TWO ROMS

All of the instructions in the 32 x 9 ROM can be modified in 8 different ways by the 8 x 8 ROM. Any instruction for example can be shifted left or right with the data out pins showing A, B, or F and the shifted result stored in the RAM or Q Register or both. If the clock is gated OFF by external logic it will be impossible to load the RAM or Q Register, and the controlled use of C_N in many instructions further raises the instruction count.

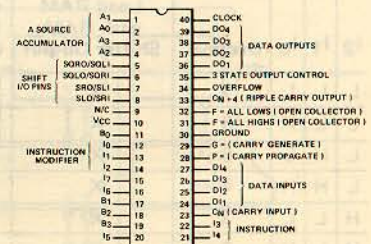
INSTRUCTIONS IN THE 32 x 9 ROM – POSITIVE LOGIC (1 = H ≈ 3 V) INTERPRETATION

ROM WORD							ALU Instruction (See Pg. 10 for Symbology)	ALU OUTPUT		TYPICAL USES
I ₇	I ₆	I ₅	I ₄	I ₃	Decimal	Octal		No Carry In (C _N = L)	With Carry In (C _N = H)	
L	L	L	L	L	0	00	LLLL + HHHH + C _N	Force 1111	Force 0000	Initialization (Force 1's or 0's)
L	L	L	L	H	1	01	AND A ₁ & B _j	A ₁ ∧ B _j	A ₁ ∧ B _j	AND A ₁ & B _j
L	L	L	H	L	2	02	AND D ₁ & B _j	D ₁ ∧ B _j	D ₁ ∧ B _j	" D ₁ & B _j
L	L	L	H	H	3	03	OR A ₁ & B _j	A ₁ ∨ B _j	A ₁ ∨ B _j	OR A ₁ & B _j
L	L	H	L	L	4	04	OR D ₁ & B _j	D ₁ ∨ B _j	D ₁ ∨ B _j	" D ₁ & B _j
L	L	H	L	H	5	05	Exclusive OR A ₁ & B _j	A ₁ ⊕ B _j	A ₁ ⊕ B _j	Exclusive Or A ₁ & B _j
L	L	H	H	L	6	06	Exclusive OR D ₁ & B _j	D ₁ ⊕ B _j	D ₁ ⊕ B _j	" D ₁ & B _j
L	L	H	H	H	7	07	$\overline{A_1} + HHHH + C_N$	$\overline{A_1} + 1111$	$\overline{A_1}$	Invert A ₁
L	H	L	L	L	8	10	$\overline{D_1} + HHHH + C_N$	$\overline{D_1} + 1111$	$\overline{D_1}$	" D ₁
L	H	L	L	H	9	11	$\overline{B_j} + HHHH + C_N$	$\overline{B_j} + 1111$	$\overline{B_j}$	" B _j
L	H	L	H	L	10	12	$\overline{Q} + HHHH + C_N$	$\overline{Q} + 1111$	\overline{Q}	" Q
L	H	L	H	H	11	13	$\overline{A_1} + LLLL + C_N$	$\overline{A_1}$	$\overline{A_1} + 0001$	2's Complement Of A ₁
L	H	H	L	L	12	14	$\overline{D_1} + LLLL + C_N$	$\overline{D_1}$	$\overline{D_1} + 0001$	" D ₁
L	H	H	L	H	13	15	$\overline{B_j} + LLLL + C_N$	$\overline{B_j}$	$\overline{B_j} + 0001$	" B _j
L	H	H	H	L	14	16	$\overline{Q} + LLLL + C_N$	\overline{Q}	$\overline{Q} + 0001$	" Q
L	H	H	H	H	15	17	A ₁ + LLLL + C _N	A ₁	A ₁ + 0001	Transfer Or Increment A ₁
H	L	L	L	L	16	20	D ₁ + LLLL + C _N	D ₁	D ₁ + 0001	" D ₁
H	L	L	L	H	17	21	B _j + LLLL + C _N	B _j	B _j + 0001	" B _j
H	L	L	H	L	18	22	Q + LLLL + C _N	Q	Q + 0001	" Q
H	L	L	H	H	19	23	A ₁ + HHHH + C _N	A ₁ + 1111	A ₁	Decrement Or Transfer A ₁
H	L	H	L	L	20	24	D ₁ + HHHH + C _N	D ₁ + 1111	D ₁	" D ₁
H	L	H	L	H	21	25	B _j + HHHH + C _N	B _j + 1111	B _j	" B _j
H	L	H	H	L	22	26	Q + HHHH + C _N	Q + 1111	Q	" Q
H	L	H	H	H	23	27	A ₁ + B _j + C _N	A ₁ + B _j	A ₁ + B _j + 0001	Add A ₁ & B _j
H	H	L	L	L	24	30	D ₁ + B _j + C _N	D ₁ + B _j	D ₁ + B _j + 0001	" D ₁ & B _j
H	H	L	L	H	25	31	A ₁ + Q + C _N	A ₁ + Q	A ₁ + Q + 0001	" A ₁ & Q
H	H	L	H	L	26	32	D ₁ + Q + C _N	D ₁ + Q	D ₁ + Q + 0001	" D ₁ & Q
H	H	L	H	H	27	33	A ₁ + $\overline{B_j} + C_N$	A ₁ - B _j - 0001	A ₁ - B _j	Subtract A ₁ & B _j
H	H	H	L	L	28	34	B _j + A ₁ + C _N	B _j - A ₁ - 0001	B _j - A ₁	" B _j & A ₁
H	H	H	L	H	29	35	D ₁ + $\overline{B_j} + C_N$	D ₁ - B _j - 0001	D ₁ - B _j	" D ₁ & B _j
H	H	H	H	L	30	36	B _j + $\overline{D_1} + C_N$	B _j - D ₁ - 0001	B _j - D ₁	" B _j & D ₁
H	H	H	H	H	31	37	D ₁ + $\overline{Q} + C_N$	D ₁ - Q - 0001	D ₁ - Q	" D ₁ & Q

INSTRUCTION MODIFIERS IN THE 8 x 8 ROM – POSITIVE LOGIC (1 = H ≈ 3 V) INTERPRETATION

Rom Word		Rom Word		Load Control		Shift Control			Data Out Control		
I ₂	I ₁	I ₀	Decimal	Load Ram B _j	Load Q	Shift Left	Shift Right	Don't Shift	A Latch	B Latch	ALU Output F
L	L	L	0	X				X			X
L	L	H	1	X				X	X		
L	H	L	2	X				X		X	
L	H	H	3	X		X					X
H	L	L	4	X			X				X
H	L	H	5	X	X	X					X
H	H	L	6	X	X		X				X
H	H	H	7		X			X			X

PIN CONFIGURATION

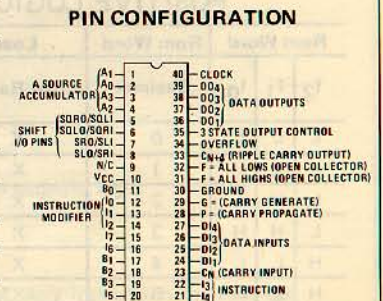


INSTRUCTIONS IN THE 32 x 9 ROM – NEGATIVE LOGIC (1 = L ≈ 0 V) INTERPRETATION

ROM WORD							ALU Instruction (See Pg.10 for Symbology)	ALU OUTPUT		TYPICAL USES
								No Carry In (C _N = H)	With Carry In (C _N = L)	
I ₇	I ₆	I ₅	I ₄	I ₃	Decimal	Octal				
L	L	L	L	L	31	37	LLLL + HHHH + C _N	Force 1111	Force 0000	Initialization (Force 1's or 0's)
L	L	L	L	H	30	36	OR A ₁ & B _j	A ₁ ∨ B _j	A ₁ ∨ B _j	OR A ₁ & B _j
L	L	L	H	L	29	35	OR D ₁ & B _j	D ₁ ∨ B _j	D ₁ ∨ B _j	" D ₁ & B _j
L	L	L	H	H	28	34	AND A ₁ & B _j	A ₁ ∧ B _j	A ₁ ∧ B _j	AND A ₁ & B _j
L	L	H	L	L	27	33	AND D ₁ & B _j	D ₁ ∧ B _j	D ₁ ∧ B _j	" D ₁ & B _j
L	L	H	L	H	26	32	Exclusive NOR A ₁ & B _j	A ₁ ⊘ B _j	A ₁ ⊘ B _j	Exclusive Nor A ₁ & B _j
L	L	H	H	L	25	31	Exclusive NOR D ₁ & B _j	D ₁ ⊘ B _j	D ₁ ⊘ B _j	" D ₁ & B _j
L	L	H	H	H	24	30	A ₁ + HHHH + C _N	A ₁	A ₁ + 0001	2's Complement Of A ₁
L	H	L	L	L	23	27	D ₁ + HHHH + C _N	D ₁	D ₁ + 0001	" D ₁
L	H	L	L	H	22	26	B _j + HHHH + C _N	B _j	B _j + 0001	" B _j
L	H	L	H	L	21	25	Q + HHHH + C _N	Q	Q + 0001	" Q
L	H	L	H	H	20	24	A ₁ + LLLL + C _N	A ₁ + 1111	A ₁	Invert A ₁
L	H	H	L	L	19	23	D ₁ + LLLL + C _N	D ₁ + 1111	D ₁	" D ₁
L	H	H	L	H	18	22	B _j + LLLL + C _N	B _j + 1111	B _j	" B _j
L	H	H	H	L	17	21	Q + LLLL + C _N	Q + 1111	Q	" Q
L	H	H	H	H	16	20	A ₁ + LLLL + C _N	A ₁ + 1111	A ₁	Decrement Or Transfer A ₁
H	L	L	L	L	15	17	D ₁ + LLLL + C _N	D ₁ + 1111	D ₁	" D ₁
H	L	L	L	H	14	16	B _j + LLLL + C _N	B _j + 1111	B _j	" B _j
H	L	L	H	L	13	15	Q + LLLL + C _N	Q + 1111	Q	" Q
H	L	L	H	H	12	14	A ₁ + HHHH + C _N	A ₁	A ₁ + 0001	Transfer Or Increment A ₁
H	L	H	L	L	11	13	D ₁ + HHHH + C _N	D ₁	D ₁ + 0001	" D ₁
H	L	H	L	H	10	12	B _j + HHHH + C _N	B _j	B _j + 0001	" B _j
H	L	H	H	L	9	11	Q + HHHH + C _N	Q	Q + 0001	" Q
H	L	H	H	H	8	10	A ₁ + B _j + C _N	A ₁ + B _j	A ₁ + B _j + 0001	Add A ₁ & B _j
H	H	L	L	L	7	07	D ₁ + B _j + C _N	D ₁ + B _j	D ₁ + B _j + 0001	" D ₁ & B _j
H	H	L	L	H	6	06	A ₁ + Q + C _N	A ₁ + Q	A ₁ + Q + 0001	" A ₁ & Q
H	H	L	H	L	5	05	D ₁ + Q + C _N	D ₁ + Q	D ₁ + Q + 0001	" D ₁ & Q
H	H	L	H	H	4	04	A ₁ + B _j + C _N	A ₁ - B _j - 0001	A ₁ - B _j	Subtract A ₁ & B _j
H	H	H	L	L	3	03	B _j + A ₁ + C _N	B _j - A ₁ - 0001	B _j - A ₁	" B _j & A ₁
H	H	H	L	H	2	02	D ₁ + B _j + C _N	D ₁ - B _j - 0001	D ₁ - B _j	" D ₁ & B _j
H	H	H	H	L	1	01	B _j + D ₁ + C _N	B _j - D ₁ - 0001	B _j - D ₁	" B _j & D ₁
H	H	H	H	H	0	00	D ₁ + Q + C _N	D ₁ - Q - 0001	D ₁ - Q	" D ₁ & Q

INSTRUCTION MODIFIERS IN THE 8 x 8 ROM – NEGATIVE LOGIC (1 = L ≈ 0 V) INTERPRETATION

ROM Word				Load Control		Shift Control			Data Out Control			
I ₂	I ₁	I ₀	Decimal	Load RAM with RAM Shifter Output	Load Q		Shift Left	Shift Right	Don't Shift	A Latch	B Latch	ALU Output F
					with Q Shifter Output	with ALU Output						
L	L	L	7	X					X			X
L	L	H	6	X					X	X		
L	H	L	5	X					X		X	
L	H	H	4	X			X					X
H	L	L	3	X				X				X
H	L	H	2	X	X		X					X
H	H	L	1	X	X			X				X
H	H	H	0			X			X			X



APPLICATION TRICKS WITH SOME INSTRUCTIONS

- 1) $A_i \wedge B_j$ or $A_i \vee B_j$ can be used as a no operation if $A_i = B_j$ and only the RAM is loaded since we are loading a register with itself.
- 2) $A_i \nabla B_j$ can be used to load LLLL if $A_i = B_j$ regardless of the state of C_N .
- 3) $A_i + B_j + C_N$ gives $A_i + A_i + C_N$ if $A_i = B_j$, forming twice "A" or twice (A + I). IF $A_i + B_j + C_N$ is shifted left with $A_i = B_j$, we get a double left shift since adding a number to itself is a multiply by 2 which in binary is a shift left.
- 4) Q can be shifted and stored back in Q during any instruction involving HLH or HHL on I_2, I_1, I_0 permitting simultaneous RAM and Q shifting. Operations involving 3 registers (A, B, and Q) are also possible since we can perform $A + B \rightarrow Q$.
- 5) $B_j + LLLL + C_N \rightarrow B_j$
 $B_j + HHHH + C_N \rightarrow B_j$ can be used as a NO OP if there is no carry
- 6) $B_j + LLLL + C_N \rightarrow B_j, A_i \rightarrow OUT$
 $B_j + HHHH + C_N \rightarrow B_j, A_i \rightarrow OUT$

Permits use of one of the B_j for a program counter (P.C) while looking at one of the A_i which might be an accumulator on the output pins. If there is a carry, $B_0 + 0001 \rightarrow B_0$ (Increment P.C)

$$A_i \rightarrow OUT \text{ (Old P.C} \rightarrow OUT)$$

Might be used to increment the program counter while having the old value of the program counter addressing memory (via the data out pins) in one microcontroller cycle.

CARRY GENERATE (\bar{G}) AND CARRY PROPAGATE (\bar{P})

A) THEORY

The \bar{G} and \bar{P} pins of the microcontroller are designed to be used with the TTL 54182/74182 look-ahead carry generator. The look-ahead techniques predict whether or not there will be a carry generated from the microcontroller and whether the input carry (C_N) will be propagated through the microcontroller, based on the input operands rather than waiting for the carry to ripple through each internal stage of the microcontroller. The add times of larger word length machines are significantly faster with these techniques.

B) ACTIVE HIGH AND ACTIVE LOW LOGIC

\bar{G} and \bar{P} are sometimes called X and Y in active HIGH (Positive) logic terminology since the generate and propagate terminology are pertinent only to active LOW (Negative) logic. The microcontroller and look-ahead carry generator will produce the correct results in both active HIGH and active LOW logic. Figure 6 shows how to hook-up the look-ahead carry generator (74182 or 74S182).

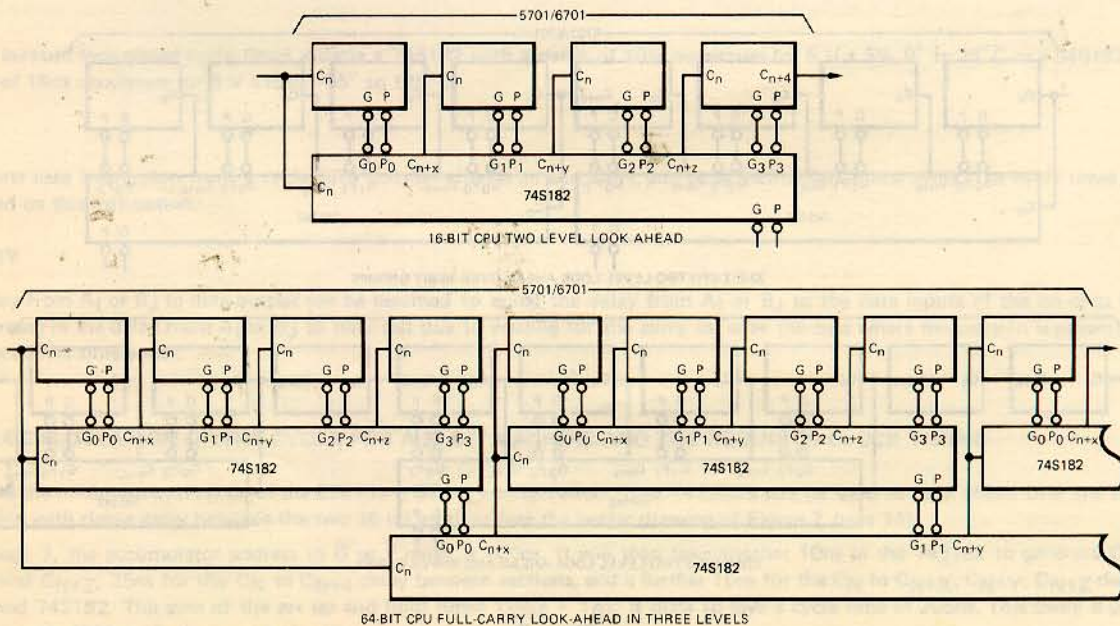


Figure 6

APPLICATION INFORMATION (Cont'd)

C) LOGIC EQUATION

In logic symbology where \cdot is a logic "AND" and $+$ is a logic "OR" and where the subscript L means a low TTL logic level ; P can be expressed as follows:

$$Y = P_L = (A'_{3H} + B'_{3H}) \cdot (A'_{2H} + B'_{2H}) \cdot (A'_{1H} + B'_{1H}) \cdot (A'_{0H} + B'_{0H})$$

This equation is interpreted to mean that propagate is a low level if A'_3 or B'_3 is high and A'_2 or B'_2 is high, and A'_1 or B'_1 is high, and A'_0 or B'_0 is high. $A'_3, A'_2, A'_1, A'_0,$ and B'_3, B'_2, B'_1, B'_0 refer to the four bit number applied at the input A' and B' of the microcontroller ALU.

In the same symbology:

$$X = G_L = (A'_{3H} \cdot B'_{3H}) + (A'_{3H} + B'_{3H}) \cdot (A'_{2H} \cdot B'_{2H}) + (A'_{3H} + B'_{3H}) \cdot (A'_{2H} + B'_{2H}) \cdot (A'_{1H} \cdot B'_{1H}) + (A'_{3H} + B'_{3H}) \cdot (A'_{2H} + B'_{2H}) \cdot (A'_{1H} + B'_{1H}) \cdot (A'_{0H} \cdot B'_{0H})$$

RIPPLE CARRY (C_{N+4})

In systems not requiring the speed of look-ahead addition the 74S182 look-ahead carry generator can be eliminated and the ripple carry C_{N+4} can be used instead. Simply tie the ripple carry output C_{N+4} of the least significant 5701/6701 package to the carry input C_N of the more significant 5701/6701 package. Figure 7 shows some examples of ripple carry and combined ripple and look-ahead carry.

The ripple carry output will be a TTL high level when a carry out occurs in the active HIGH (Positive) logic convention. The ripple carry output will be a TTL low level when a carry out occurs in the active LOW (Negative) logic convention. The ripple carry is not gated off during logic operations since the conditions defining whether it should be off or on are not the same in positive and negative logic.

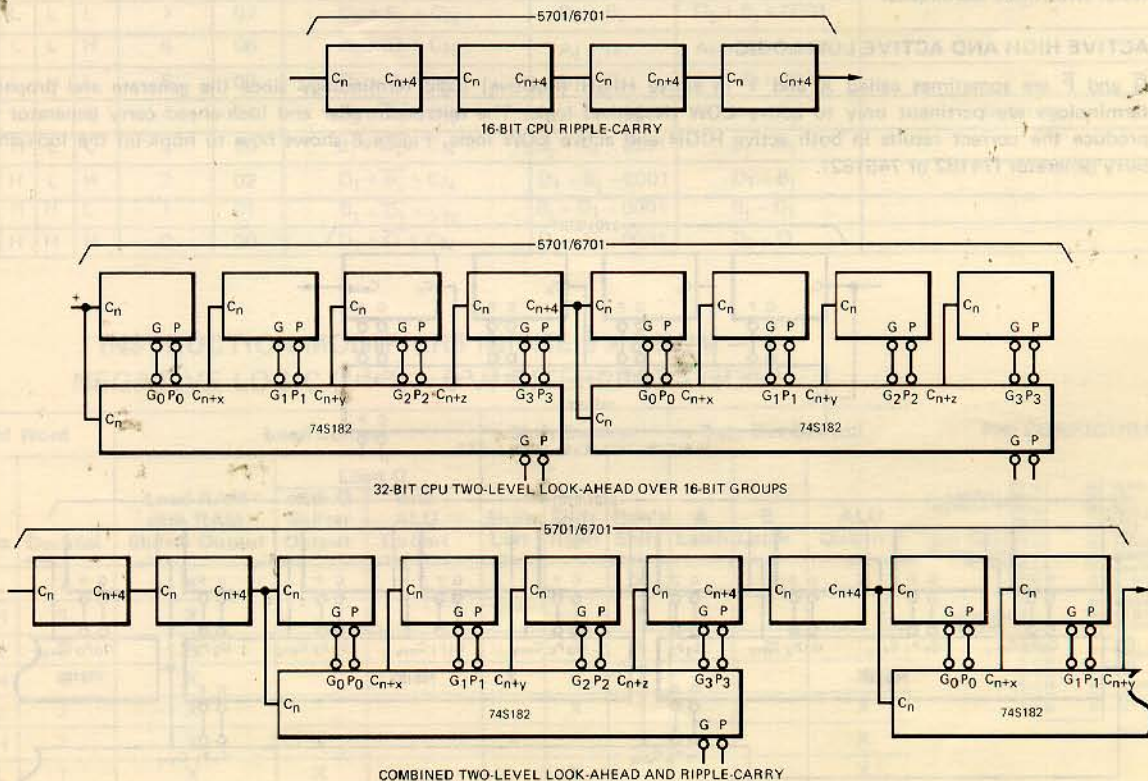


Figure 7

WORST CASE MINIMUM CYCLE TIMES OVER THE V_{CC} AND TEMPERATURE RANGE FOR CPU'S LARGER THAN 4 BITS

CPU Word Length (# Bits)	Minimum Cycle with Ripple Carry (ns)	6701 Minimum Cycle* with Look Ahead		Minimum Cycle with Ripple Carry (ns)	5701 Minimum Cycle* with Look Ahead	
		# Look Ahead 74S182s	Cycle Time (ns)		# Look Ahead 74S182s	Cycle Time (ns)
4	175			230		
8	175			230		
		1	175		1	230
12	195			255		
		1	175		1	230
16	220			285		
		1	175		1	230
24	270			345		
		1	220		1	290
32	320			405		
		1	270		1	350
		2	205		2	275
48	420			525		
		1	370		1	470
		2	305		2	395
		3	240		3	320
64	520			645		
		1	470		1	590
		2	405		2	515
		3	340		3	440
		4	275		4	365
		5	190	5	260	

*The minimum look-ahead cycle times assume a 74S182 with a delay of 10ns maximum for $5V \pm 5\%$, 0° to $75^\circ C$ or a 54S182 with a delay of 15ns maximum for $5V \pm 10\%$, -55° to $125^\circ C$.

NOTE

The worst case instruction from a cycle time consideration is an add, shift, and store instruction in one cycle. The cycle times above are based on this instruction.

THEORY

The delay from A_1 or B_j to data output can be assumed to equal the delay from A_1 or B_j to the data inputs of the on chip RAM. The increase in the delay from A_1 or B_j to data out due to waiting for the carry in, over the case where the carry in is present early is a direct cycle time adder.

SAMPLE CALCULATION OF THE CYCLE FOR A 32 BIT MACHINE AND THE REQUIRED CLOCK TIMING

Calculate the minimum cycle time of the 6701 in a 32 bit configuration. Two 74S182's will be used to look ahead over the two 16 bit section with ripple carry between the two 16 bit sections (see the center drawing of Figure 7 page 14).

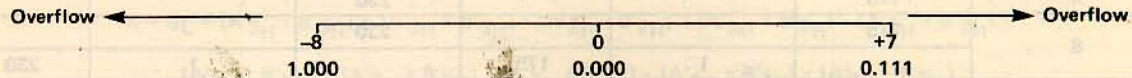
From page 7, the accumulator address to \bar{G} or \bar{P} delay is 100ns. It will then take another 10ns in the 74S182 to generate C_{N+X} , C_{N+Y} , and C_{N+Z} , 25ns for the C_N to C_{N+4} delay between sections, and a further 10ns for the C_N to C_{N+X} , C_{N+Y} , C_{N+Z} delay of the second 74S182. The sum of the set up and hold times $T_{CNS} + T_{BH}$ is 60ns to give a cycle time of 205ns. This delay is greater than the single 6701 cycle time and is the governing cycle time.

APPLICATION INFORMATION (Cont'd)

OVERFLOW

THEORY

When the result of a binary addition or subtraction requires more bits than the arithmetic unit can accommodate overflow can occur. For example, consider a 4-bit system (3 bits + sign) where the most significant bit is defined on a zero for positive numbers and a one for negative numbers. This system has a maximum value of 7 and a minimum value of -8 as indicated on the line graph below:



If we list all the addition and subtraction conditions which will give an answer not in the number system (beyond the ends of the line graph above), four conditions result in overflow and are listed below with examples:

OVERFLOW CONDITIONS

1. ADD TWO LARGE POSITIVE NUMBERS

7 plus 7 = 14, which is larger than our largest number (+7) on the line graph, therefore, overflow occurs
2 plus 1 = 3, which is smaller than +7, therefore, no overflow

2. ADD TWO LARGE NEGATIVE NUMBERS

-7 plus -7 = -14, which is beyond the end of the line graph, therefore, overflow occurs.
-2 plus -1 = -3, no overflow since -3 is within the bounds of the number system.

3. SUBTRACT A LARGE NEGATIVE NUMBER FROM A LARGE POSITIVE NUMBER

-7 minus +7 = -14, overflow occurs
-2 minus +1 = -3, no overflow

4. SUBTRACT A LARGE POSITIVE NUMBER FROM A LARGE NEGATIVE NUMBER

+7 minus -7 = +14, overflow occurs
+2 minus -1 = +3, no overflow

OVERFLOW PIN

A conventional binary number system has the most significant bit defined as the sign bit (0 is a positive number, 1 is a negative number by definition) and negative numbers are represented in a 2's complement form (the 2's complement of a number can be formed by inverting the number and adding one, for example, 0101 = +5, -5 = 1010 plus 1 or 1011).

The four overflow conditions shown above can be detected by exclusive ORing the carry-in and carry-out of the sign bit. If the carry's disagree, overflow has occurred. The overflow output will only be meaningful on the most significant 5701/6701 package in systems larger than four bits. Since the overflow is implemented with an exclusive nor gate the 5701/6701 will also give overflow outputs during logic as well as arithmetic operations requiring that external logic decide when the overflow output is meaningful. The overflow pin will be low when overflow occurs.

HARDWARE MULTIPLY AND DIVIDE

I. GENERAL DISCUSSION

The microcontroller's capability of adding and shifting or subtracting and shifting within one microinstruction cycle, and having the accumulator extension register Q on chip permits fast microprogrammed multiply and divide. Less than 20 cycles are required for a 16 bit multiply or divide.

Figure 8 shows how to connect the microcontroller for a 16 bit multiply/divide. The least significant bit of the RAM shifter is shifted into the most significant bit of Q.

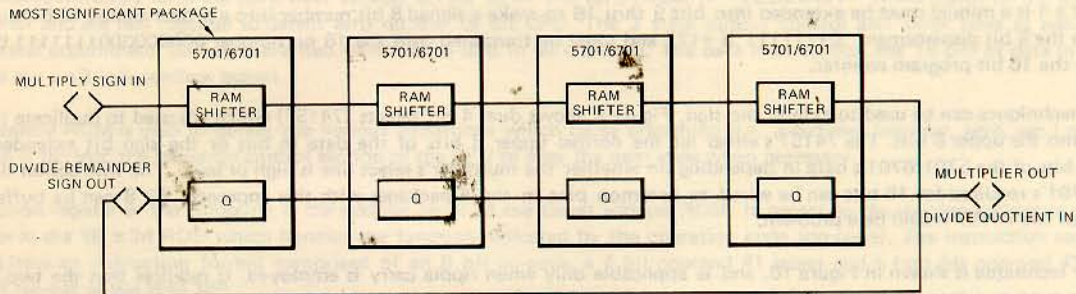


Figure 8

II. HARDWARE MULTIPLY – EXAMPLE FOR A 16 BY 16 MULTIPLY

A. INITIAL CONDITIONS

Put the multiplier in Q, multiplicand on the RAM A output, and clear the RAM B output to zero.

B. FINAL CONDITIONS

The most significant 16 bit half of the 32 bit product will be in RAM B and the least significant 16 bit half in Q. The multiplicand in A is undisturbed.

C. THEORY

A 16 by 16 multiply in the microcontroller requires 16 reduction steps. Each reduction step consists of an add ($A+B$), if the least significant bit or the multiplier emerging from Q is a one, or a transfer ($0+B$) if the least significant bit of Q is a zero, and then a right shift of the result into B and Q.

D. TWO'S COMPLEMENT

Two's complement multiplication requires sign bit manipulation and an additional correction cycle for negative multipliers. The sign inserted into the most significant bit of B during the reduction cycles is the exclusive OR of overflow and the most significant bit of the ALU output. After 16 reduction cycles, an additional correction cycle is required if the multiplier was negative. The correction cycle consists of subtracting the multiplicand from the product without shifting.

III. HARDWARE DIVIDE

Example for a 16 bit divisor and 32 bit dividend giving a 16 bit quotient and 16 bit remainder.

A. INITIALIZATION

Put the divisor on the RAM A output, and the dividend's most significant half on the RAM B output and the least significant half in Q.

B. FINAL CONDITION

The 16 bit quotient is on the RAM B output and the 16 bit remainder is in Q. The divisor is in the RAM A output undisturbed.

C. THEORY

A non-restoring two's complement division requires 16 reduction steps for negative quotients. In non-restoring division, the divisor is successively subtracted from the dividend and the result shifted left until the remainder changes sign. At this point the divisor has been subtracted one too many times and is added back until the original sign is restored.

Negative quotients are generated in one's complement notation (the one's complement of a number is its inverse). The two correction cycles provide a remainder correction if required so that the remainder sign is the same as the original dividend and quotient correction from one's complement to two's complement form for negative quotients.

Each of the 16 reduction cycles consists of a subtract ($B-A$) or an add ($A+B$) operation plus a left shift of the result. The most significant bit of Q propagates into the least significant bit of B and the quotient bit is shifted into the least significant vacated position of Q. The case of zero remainder will not require quotient correction but may require sign correction of the remainder.

SIGN EXTENSION

A common operation in computers is the calculation of an effective address. In a 16 bit computer, for example, we may have program counter relative addressing, where the addition of an 8 bit displacement to a 16 bit program counter is the effective memory address. This 8 bit displacement is a signed displacement with a value between +127 and -128.

In using the 5701/6701 to calculate an effective address, the displacement will be brought into the unit with the data in pins, and added to the program counter which is stored in one of the internal registers. The problem arises when adding an 8 bit number to a 16 bit number in that the upper 8 bits of the displacement must be ignored and the sign of the 8 bit displacement (a 0 in bit 8 is a plus and a 1 is a minus) must be extended into bits 9 thru 16 to make a signed 8 bit number into a proper signed 16 bit number. For example the 8 bit displacement 01111111 is +127 and must be translated into the 16 bit number 0000000011111111 before adding it to the 16 bit program counter.

Several techniques can be used to extend the sign. Figure 9 shows dual 4 bit selects (74157) which are used to duplicate the sign bit (bit 8) into the upper 8 bits. The 74157's either let the normal upper 8 bits of the data in bus or the sign bit extended into the upper 8 bits of the 5701/6701's data in depending on whether the multiplier's select line is high or low. The instruction lines of the 5701/6701's required for 16 bits can be wired to common pins in the 4 packages with this approach. Bit 8 can be buffered if the loading of the 74157 would be a problem.

Another technique is shown in Figure 10, and is applicable only when ripple carry is employed. It requires that the two low order packages execute a data in plus register operation and that the 2 high order packages execute a LLLL plus HHHH plus C_N instruction. If C_N is low and the 7451 is in the sign extend mode, then highs will be forced into the upper two packages. If C_N is a high and the 7451 is in the extend sign mode, then lows will be forced into the high order packages. When the sign is not to be extended the 7451 will pass the C_{N+4} ripple carry into C_N . This method requires that the lower two 5701/6701's execute a different instruction than the upper two 5701/6701's. This particular case will require changing the state of two of the 8 instruction lines which go into the upper two 5701/6701's, namely the I_6 and I_7 lines. Note that in the instruction on page 10, these instructions were purposely mapped so that few I_N lines must be changed to go from LLLL + HHHH + C_N to $D_I + B_J + C_N$.

A third alternative is to microprogram the sign extension with off chip ROMs by masking or forcing the upper 8 bits.

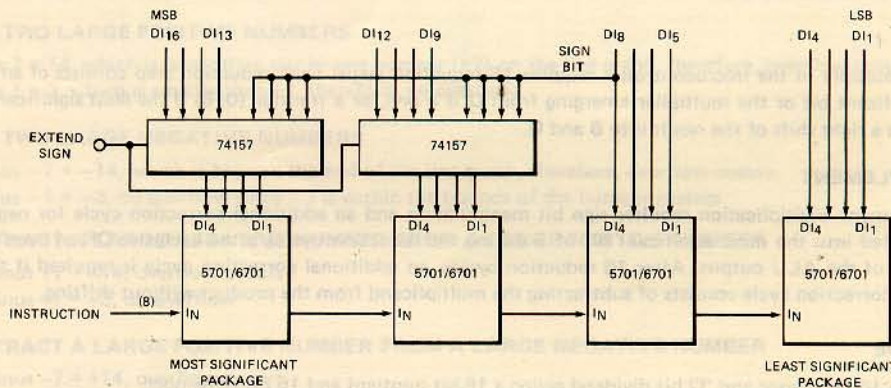


Figure 9

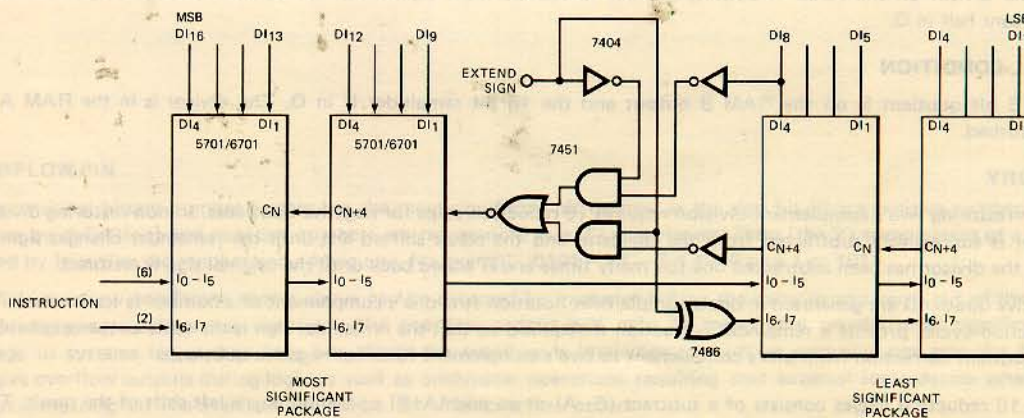


Figure 10

MICROPROGRAMMED EMULATION

Figure 11 shows a 16 bit microprogrammed CPU built from 4 microcontrollers, a few ROMs and some TTL. This CPU performs about the same function as most 16 bit minicomputer's CPU which take 110 to 225 TTL MSI packages, and require one or two 15" x 15" printed circuit boards. The CPU of Figure 11 requires 18 packages and will fit on 5" x 7" board.

HOW IT WORKS

The four microcontrollers form the data flow section of the CPU. The zero detect (F = LLLL and F = HHHH) pins of the 6701 are open collector and are tied together to detect zero on the 16 bit CPU. The 3 state output control and clock lines as well as the instruction and accumulator addresses are tied to the same pins in all 4 6701's. The data flow section has 16 bits of data in and 16 bits of data out on 2 independent busses.

A programmable ROM is used to detect the various conditions which cause branching (i.e., overflow, negative, zero, etc.) and this information is fed into the sequence control section of the CPU to alter the next state when necessary.

The instruction register of the computer is the address input to the target address ROM. It points to the starting address of a group of addresses in the 1K x 24 ROM which handles the function indicated by the operation code (op-code). The instruction register is assumed to have an instruction format comprised of an 8 bit op-code, a 4 bit operand #1 select and a four bit operand #2 select like the RR format of the IBM 360.

For example, an instruction register containing

	Op Code	File	File
might mean add file 3 to file 5 and store the	00011001	0101	0011

result in file 5. The add assembly language instruction might point to decimal address 25 (0011001) in the target address ROM which would then output the starting address of the sequence in the emulation ROM for performing addition. A typical assembly language instruction might take 4 to 10 words in the 1K x 24 ROM.

The microinstruction cycle time of the 16 bit CPU shown will be in the 350ns range. Since the microcontroller executes several operations in one cycle this CPU will execute instructions faster than most 16 bit minicomputers if semiconductor RAM is used.

The sequence of operations is programmed into the 1K x 24 for each instruction. A typical sequence for a memory reference instruction is shown below:

1. Calculate the effective address and latch it into the memory address register.
2. Fetch the contents of the memory location and latch it into the instruction register. If the memory has a 160ns or less access time, this step will not cost a microcontroller cycle.
3. Execute the instruction which is now present in the instruction register. Each clock pulse will output the required instruction to the microcontroller. When the sequence is ended the 1K x 24 ROM will disable the dual 10 bit select multiplexer gate which has been forming its next address and allow the next op-code in via the starting address ROM.
4. Increment the program counter and fetch the next instruction. A typical memory reference instruction will take 3 microcycles which includes most addressing modes, except indirect addressing. A 160ns access time, or less access time memory is required for emulation in 3 microcycles.

EMULATION

The advantage of the microprogrammed CPU described above is that it readily permits a machine designed with a new technology to be software compatible with an existing machine. The hardware technique is called emulation. Emulation protects the manufacturer's and customer's large investment in software.

The CPU configuration shown in Figure 11 can be used in many design applications where microprogrammed control might be thought of as too complicated, now that the package count has been reduced by the LSI microcontroller.

APPLICATION INFORMATION (Cont'd)

16 BIT MICROPROGRAMMED CPU

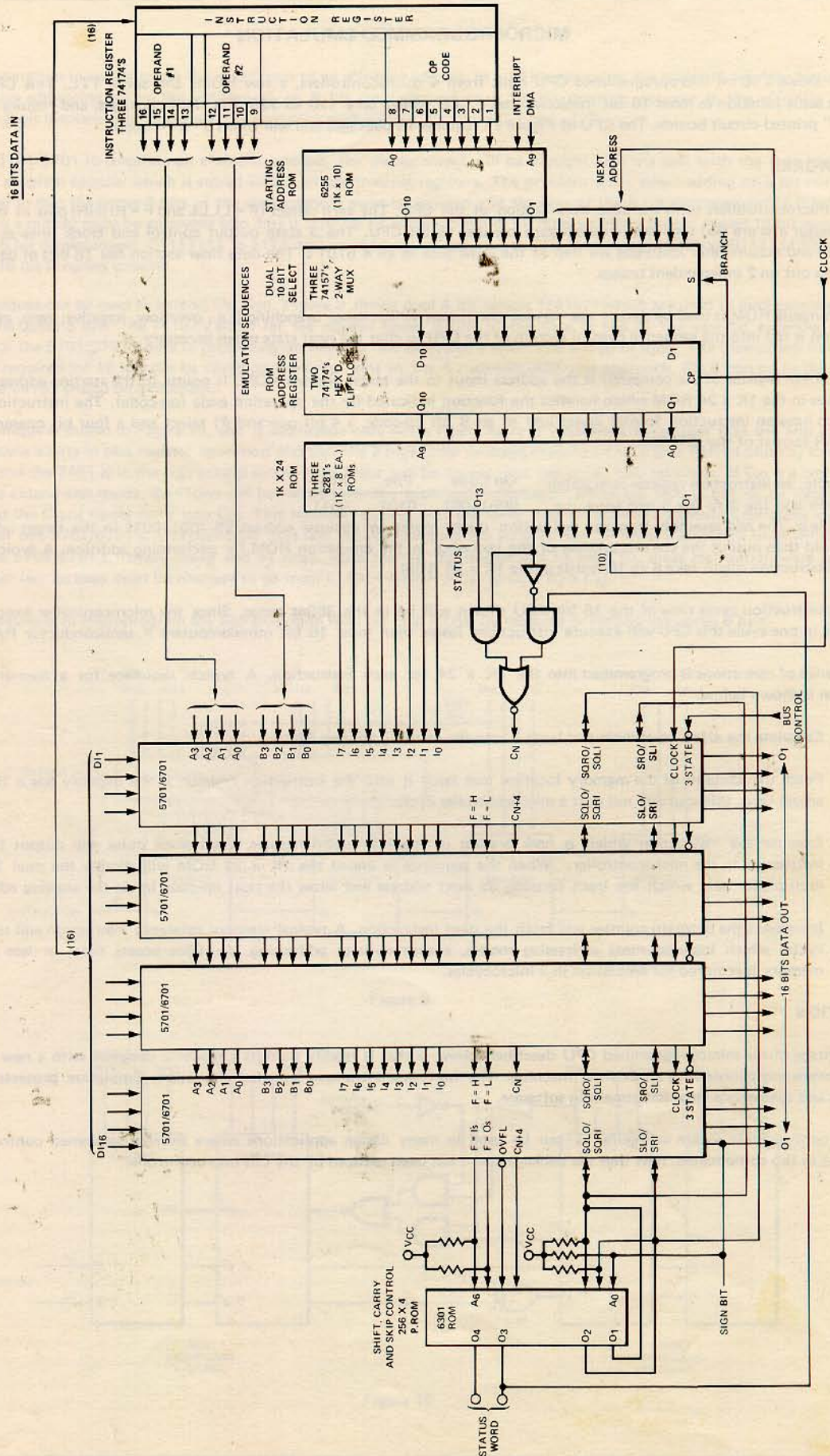


Figure 11