4-BIT EXPANDABLE BIPOLAR MICROCONTROLLER

5701/6701

	MILITARY	COMMERCIAL
5701	X	
6701		X

Monolithic Memories
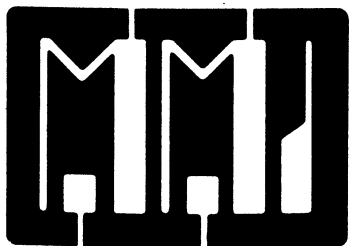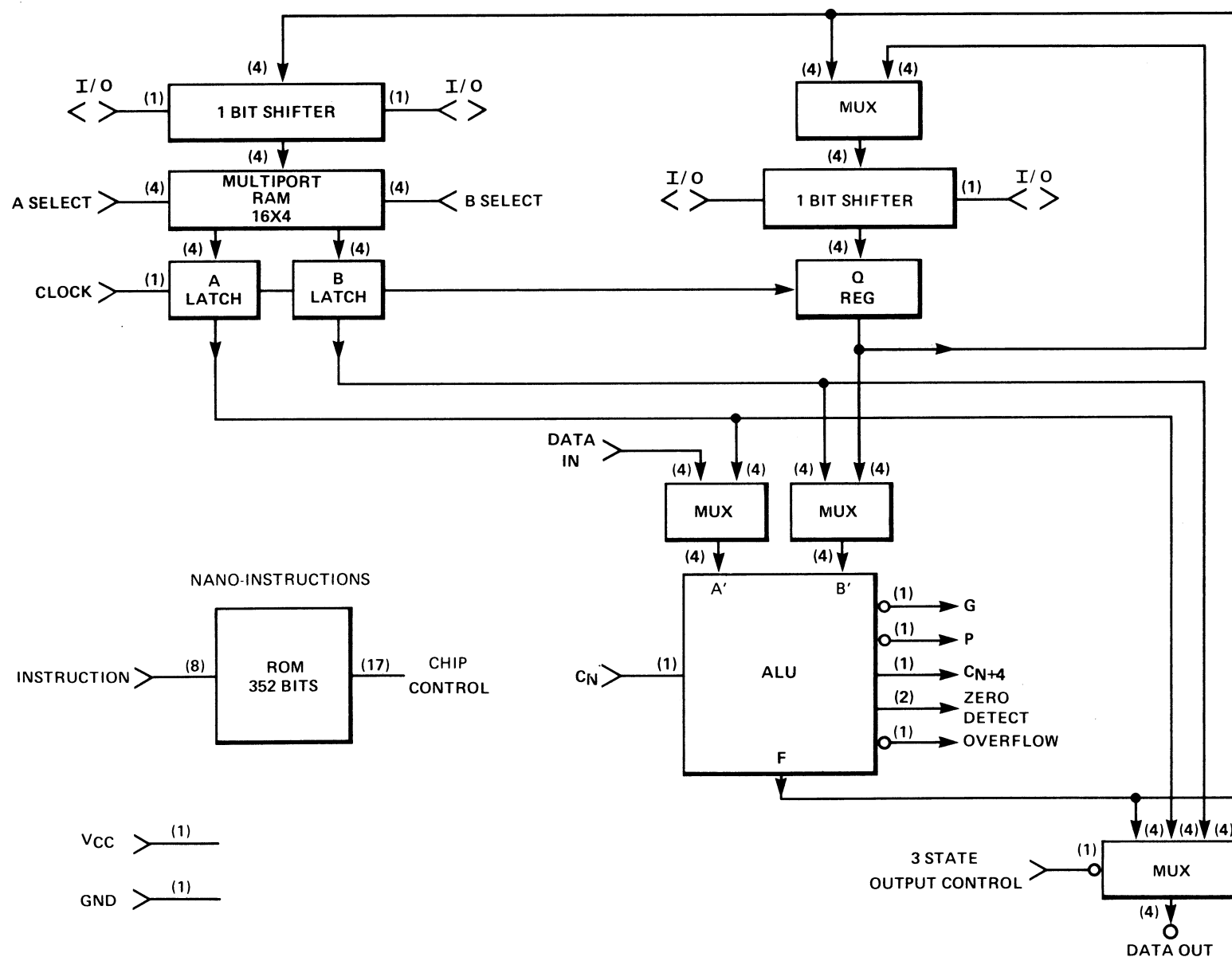INCORPORATED

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.
- 200 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

NOTE: THE NUMBERS IN PARENTHESES ARE THE NUMBER OF SIGNAL LINES.

MAY 1974

I. INTRODUCTION

A) GENERAL DESCRIPTION

The microcontroller 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 1000 gate complexity bipolar LSI device.

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 microcontroller 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 give the microcontroller the type of C.P.U. usually found only in high perform-ance 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 microcontrollers data in pins.

B) TTL EQUIVALENT

The microcontroller 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.

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	150*	450	.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		2Q	5		1032		6.63

*NOTE: 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 150 gates.

TABLE 1

C) PROCESS AND PACKAGING

The microcontroller 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

The microcontroller can execute one instruction every 200ns (250ns for the 5701). The instructions are more complex than normal micro-instructions permitting multiple operations in one cycle without timing problems.

II. OPERATION

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 lines 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.

The microcontroller is offered with a standard instruction set but the on chip ROM's can be custom coded for special customer requirements. (Contact factory for details.)

C) MULTIPORT RAM (UPPER LEFT CORNER)

A 16 word by 4 bit multiple port memory is used to fetch two operands at the same time since the RAM is double decoded (4 A address pins and 4B address pins). 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.

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 microcontroller without tying up the data out bus.

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 transfer straight through into the Q register or be shifted 1 bit left or right before being entered into the Q register. The four 3 to 1 multiplexer above the RAM function in the same manner. 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 shift on itself and be loaded from the ALU bus while any instruction is being executed. Its shift control pins, are in common with RAM shift controls permitting RAM shifting into Q and vice versa in one cycle. 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 when the clock goes low (assuming it is enabled by the ROM) and transferred from the master to the slave when the clock goes high.

EXPANDABLE 4 BIT SLICE — 40 PIN PKG.

LEGEND

⊸———▶ = PACKAGE INPUT
⊸◀——— = PACKAGE OUTPUT
⊸◀——▶ = PACKAGE INPUT AND OUTPUT

SHIFT LEFT OUT (SLO)
SHIFT RIGHT IN (SRI)

SHIFT RIGHT OUT (SRO)
SHIFT LEFT IN (SLI)

S_3

SHIFT Q RIGHT IN (SQRI)

SHIFT Q LEFT OUT (SQLO)

S_1

SHIFT Q RIGHT OUT (SQRO)

SHIFT Q LEFT IN (SQLI)

S_3

S_1

S_1
S_2
S_3

| 3 TO 1 MUX | 3 TO 1 MUX | 3 TO 1 MUX | 3 TO 1 MUX |

TRANSFER
SHIFT LEFT
SHIFT RIGHT

S_1
S_2
S_3

| 3 TO 1 MUX | 3 TO 1 MUX | 3 TO 1 MUX | 3 TO 1 MUX |

WE D_4 D_3 D_2 D_1

SOURCE A
A_3
A_2
A_1
A_0

64 BIT (16 × 4) MULTIPORT RAM

B_3
B_2
B_1
B_0

SOURCE B AND DESTINATION

O_{4A} O_{1A} O_{4B} O_{1B}

D_4 D_3 D_2 D_1

S_5

LOAD

E   "Q" REGISTER (MASTER/SLAVE)

Q_4 Q_1

S_4

CLOCK

A LATCH   B LATCH

FUNCTION S_9 S_{10}
OUTPUT = H, NEITHER SELECTED   L   L
SELECT A   L   H
SELECT DATA IN   H   L
"WIRE NOR" OF DATA IN & A   H   H

DATA IN
DI_1
DI_4

LET THRU B OR Q
OR ALL HIGHS

S_9
S_{10}

DUAL 4 BIT SELECT

DUAL 4 BIT SELECT

S_{12}
S_{13}

FUNCTION LIKE S_9 & S_{10}

LET THRU DATA IN
OR A OR ALL HIGHS

S_{11}

TRUE/ COMP.

S_{14}

TRUE/ COMP.

32 × 9 ROM

I_7
I_6
I_5
I_4
I_3

S_9 SELECT A
S_{10} SELECT DI
S_{11} TRUE/COMPL.
S_{12} SELECT B
S_{13} SELECT Q
S_{14} TRUE/COMPL.
S_{15} ALU (ADD/TRANSFER)
S_{16} ALU (LOGIC INVERT)
S_{17} ALU (KILL CARRY)

ARITH. OR LOGIC
S_{17}
S_{15}
S_{16}
C_N

A^1 ARITHMETIC LOGIC UNIT B^1

F C_{N+4}

G
P
F = ALL HIGH
F = ALL LOW
OVERFLOW

INSTRUCTION

8 × 8 ROM

I_2
I_1
I_0

S_1 SHIFT LEFT
S_2 NO SHIFT
S_3 SHIFT RIGHT
S_4 LOAD RAM
S_5 LOAD Q
S_6 OUTPUT A LATCH
S_7 OUTPUT B LATCH
S_8 OUTPUT F

AND, OR, EXCLUSIVE OR,
FORCE, INVERT, ADD,
SUBTRACT, INCREMENT,
DECREMENT, TRANSFER,
FORM 2'S COMPLEMENT

C_{N+4}

LET THRU A,B,F
S_6
S_7
S_8

3 STATE OUTPUT CONTROL

| 3 TO 1 MUX | 3 TO 1 MUX | 3 TO 1 MUX | 3 TO 1 MUX |

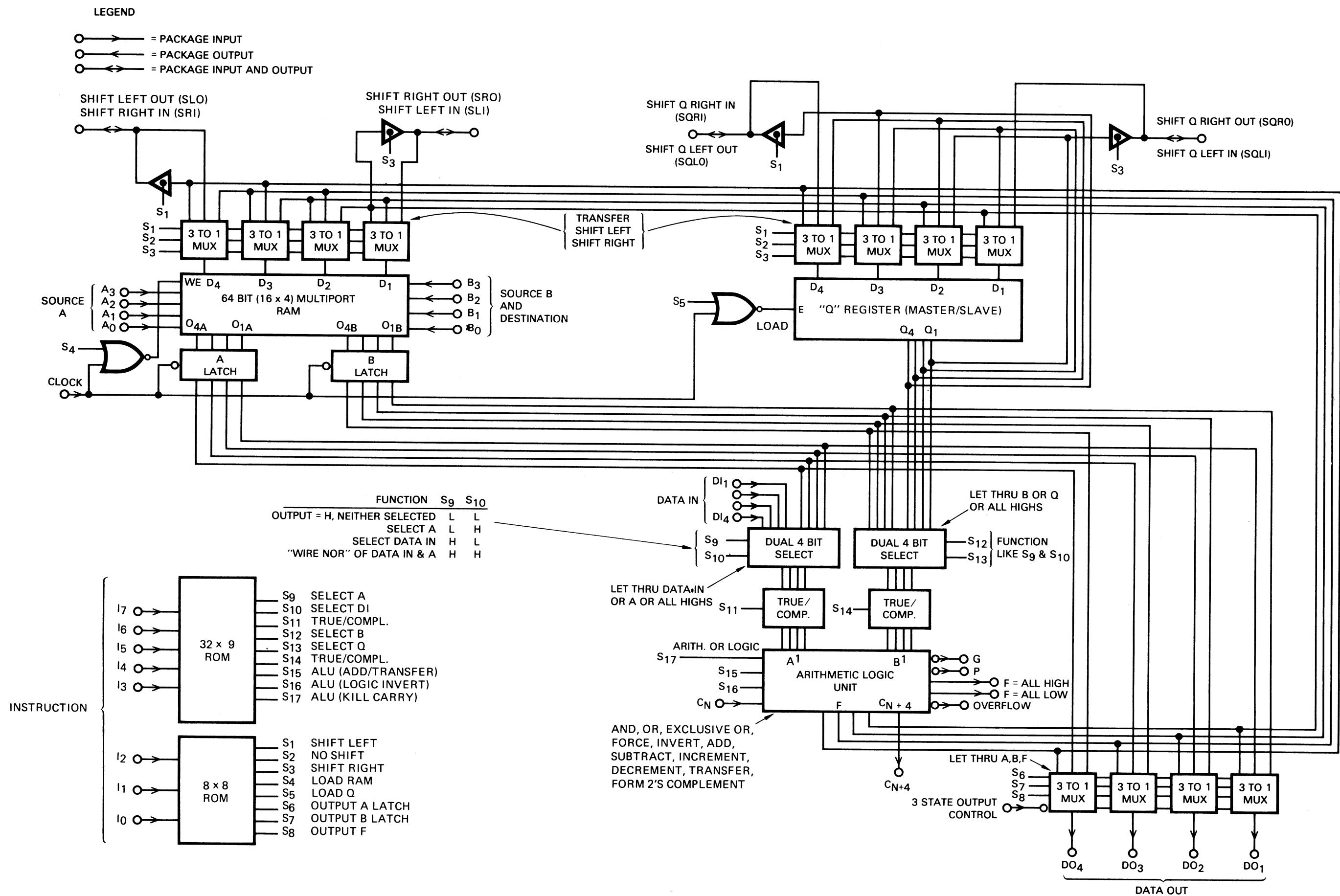DO_4 DO_3 DO_2 DO_1

DATA OUT

FIGURE 1

3

III. STATE OF THE ART ALU'S

A comparsion of the 6701 with commercially available ALU's is shown in table 2.

TABLE 2 – 4 BIT ARITHMETIC UNITS

	Arithmetic Logic Unit 74S181	Monolithic Accumulator 74S281	Microcontroller 6701
Year Introduced	1972	1973	1974
Gate Complexity	75	125	1000
Package Size	16 Pin	24 Pin	40 Pin
Number of Accumulators	0	1	16
Multiport Accumulators	No	No	Yes
Number of Useful Arithmetic Instructions	4	11	21
Number of Useful Logic Instructions	10	11	15
One Bit Bidirectional Shift Capability	No	Yes	Yes
Interface to 74S182 Look Ahead Carry Generator	Yes	Yes	Yes
Simultaneous Entry of Two Operands	Yes	No	Yes
Three State Outputs	No	No	Yes
Zero Detect	Yes	No	Yes
Overflow Detection	No	No	Yes
Accumulator Extension Register	No	No	Yes
Accumulator Extension Register One Bit Shift Capability	No	No	Yes
Operate on 3 Registers in One Cycle	NA	No	Yes
Symmetrical ALU (i.e. A minus B and B minus A)	No	Yes	Yes
Number of ALU input ports	2	2	4
Number of ALU output ports	1	1	3
Power Dissipation (TYP)	600 mW	750 mW	900 mW
Power (mW) per Gate	8	6	.9
ALU Time Required to Add Two 4 Bit Operands and Store the Result in an Accumulator Inside the ALU. A CPU with 3 to 16 Registers is Assumed	NA	205 ns (Max @ 5.0 V, 25°C) Requires Off Chip 35 ns 64 Bit (16 x 4) RAM	200 ns (Max @ 5.0 V ± 5% 0 to 75°C)
Number of Microcycles Required to Add Two 4 Bit Operands, Store the Result in an Accumulator and Look at the Result	NA	4	1

IV. CPU'S LARGER THAN 4 BITS

CPU's larger than 4 bits will have to wait for the carry generated by an off chip look ahead 74S182 or the ripple carry generated by the lower order package's C_{N+4}. As a result the minimum cycle time of the 5701/6701 will increase. When drawing timing diagram for these CPU's the amount of time the clock is high (T_{CS} of Figure 5) should be increased to permit a longer time for the ALU to stabilize based on the carry from lower order packages before bringing the clock low and writing into the RAM. The required clock low time (T_{CW} of Figure 5) does not have to be increased for CPU's larger than 4 bits. Page 14 shows the minimum cycles for larger machines.

IV. PACKAGE PINS

Figure 2 shows a listing of the 40 pin I/O. The microcontroller will use a standard 40 lead ceramic dip approximately 2.00 inches long and .60 inches wide. See Figure 3 for the package details. NOTE THAT V_{CC} 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.
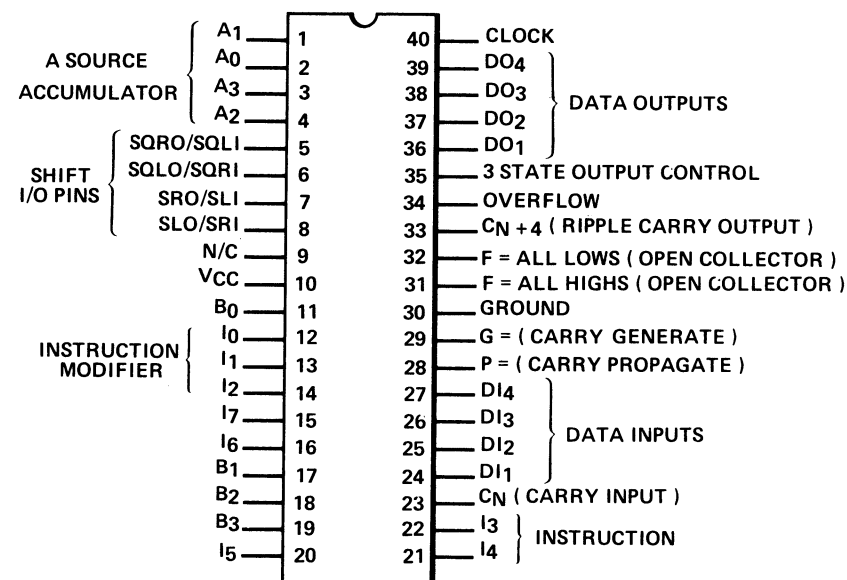
PIN CONFIGURATION

	Pin		Pin	
A SOURCE ACCUMULATOR	A_1 — 1		40 — CLOCK	
	A_0 — 2		39 — DO_4	
	A_3 — 3		38 — DO_3	DATA OUTPUTS
	A_2 — 4		37 — DO_2	
SHIFT I/O PINS	SQRO/SQLI — 5		36 — DO_1	
	SQLO/SQRI — 6		35 — 3 STATE OUTPUT CONTROL	
	SRO/SLI — 7		34 — OVERFLOW	
	SLO/SRI — 8		33 — C_{N+4} (RIPPLE CARRY OUTPUT)	
	N/C — 9		32 — F = ALL LOWS (OPEN COLLECTOR)	
	V_{CC} — 10		31 — F = ALL HIGHS (OPEN COLLECTOR)	
INSTRUCTION MODIFIER	B_0 — 11		30 — GROUND	
	I_0 — 12		29 — G = (CARRY GENERATE)	
	I_1 — 13		28 — P = (CARRY PROPAGATE)	
	I_2 — 14		27 — DI_4	
	I_7 — 15		26 — DI_3	DATA INPUTS
	I_6 — 16		25 — DI_2	
	B_1 — 17		24 — DI_1	
	B_2 — 18		23 — C_N (CARRY INPUT)	
	B_3 — 19		22 — I_3	INSTRUCTION
	I_5 — 20		21 — I_4	

Figure 2.

PACKAGE OUTLINE

40 Pin Ceramic (Side Braze)

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

Θ_{JC} (thermal resistance from junction to case with freon as a heat sink) $\approx 20^{\circ}$ C/watt
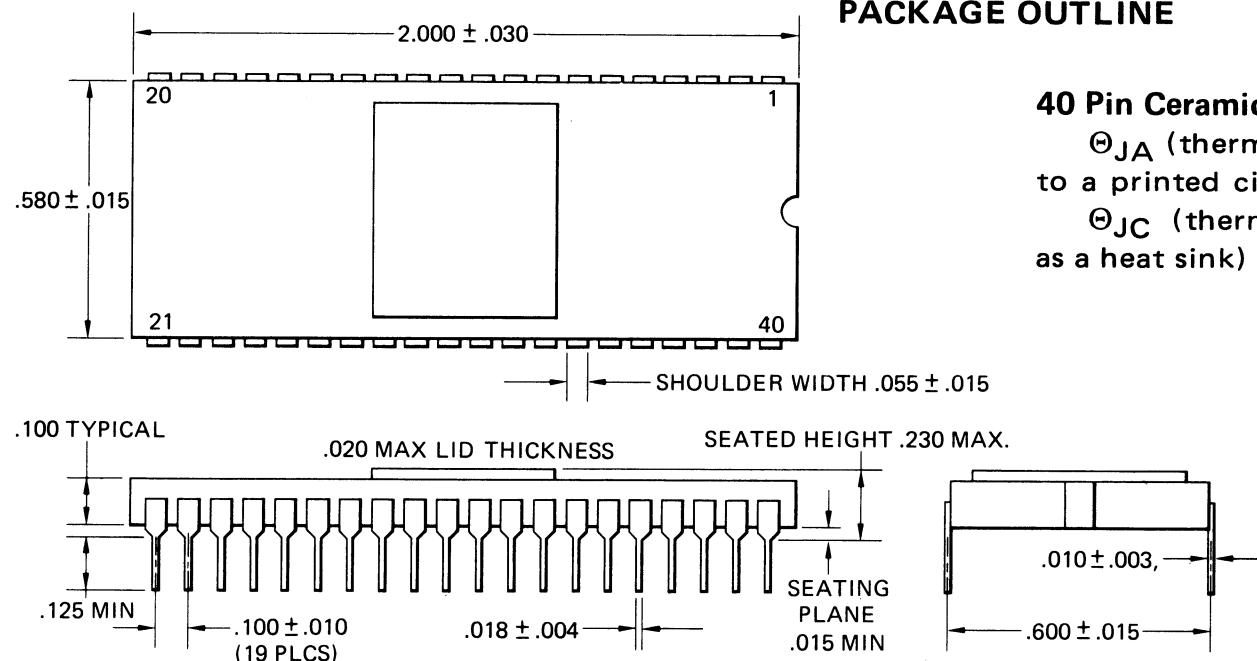
Figure 3.

BURN-IN CIRCUIT

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.
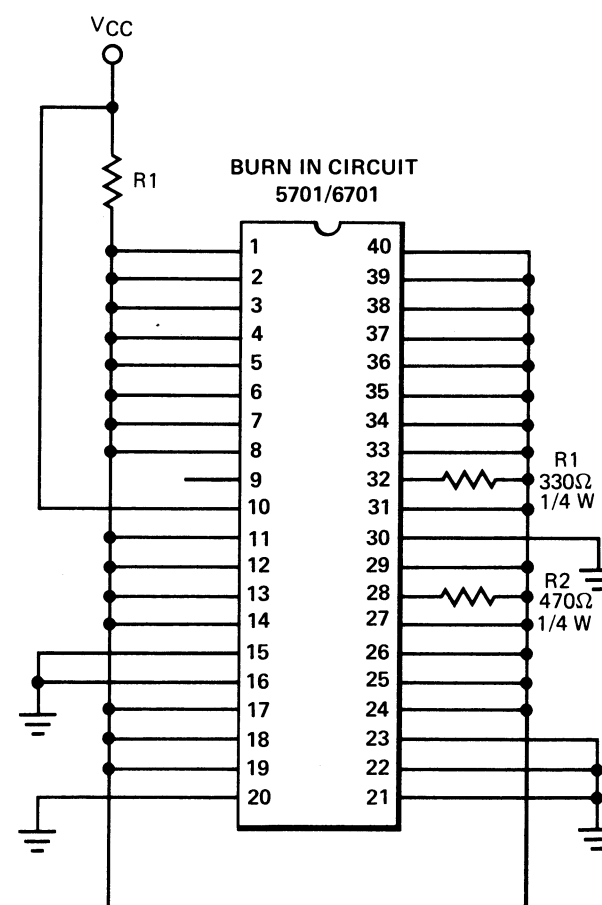

Figure 4.

ELECTRICAL PARAMETERS

ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V_{CC}	−0.5 to 7 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 V ±5% in a free air temperature of 0 to 75°C; all limits for the 5701 are guaranteed for 5 V ±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} = Max, V_F = .45 V			−250			−250	μA
	Clock or 3 State Control	"			−250			−250	μA
	Shift I/O & Data In	"			−0.80			−0.80	mA
	C_N	"			−4.80			−4.80	mA
I_R Input Leakage Current	Any A, B, or I	V_{CC} = Max, V_R = 2.40 V			40			25	μA
	Clock or 3 State Control	"			40			25	μA
	Shift I/O	Used as an Input			100			100	μA
	Data Inputs	"			20			20	μA
	C_N	"			120			120	
I_{RB} Input Leakage Current	Any A, B, or I	V_{CC} = Max, V_{RB} = 5.50 V			1.0			1.0	mA
	Clock or 3 State Control	"			1.0			1.0	mA
	Shift I/O	"			1.0			1.0	mA
	Data Input	"			1.0			1.0	mA
	C_N	"			1.0			1.0	mA
V_{OL} Low Level Output Voltage	G, C_{N+4}, F = All High, F = All Low	V_{CC} = Min, I_{OL} = 16 mA		.35	0.50		.35	0.50	V
	P, Overflow	V_{CC} = Min, I_{OL} = 10 mA		.35	0.50		.35	0.50	V
	Shift I/O	V_{CC} = Min, I_{OL} = 6 mA Used as an Output		.35	0.50		.35	0.50	V
I_{CC} Power Supply Current	V_{CC}, Grd	All inputs ground (worst case), All Outputs Open, Shift I/O Open V_{CC} = 5.00 V and Temp = Max		215	250		215	280	mA
		All Inputs ground (worst case), All Outputs Open, Shift I/O Open V_{CC} = 5.00 V and Temp = Min		230	280		230	250	mA
V_{IL} Low Level Input Voltage	All Inputs and Shift I/O	V_{CC} = 5.00 V			0.80			0.80	V
V_{IH} High Level Input Voltage	All Inputs and Shift I/O	V_{CC} = 5.00 V	2.0			2.0			V
V_{IC} Input Clamp Voltage	All Inputs and Shift I/O	V_{CC} = Min, I_{IC} = −5.0 mA		−1.0	−1.5		−1.0	−1.5	V
I_{CEX} Output Leakage Current	All Outputs and Shift I/O	V_{CC} = Max, V_{CEX} = 2.40 V, High Stored or Disabled			100			100	μA
		V_{CC} = Max, V_{CEX} = 0.45 V Disabled			−100			−100	μA
I_{CEXB} Output Leakage Current	All Outputs and Shift I/O	V_{CC} = Max = V_{CEXB} High Stored or Disabled			1			1	mA
I_{SC} Output Short Circuit Current	DO_1, DO_2, DO_3, DO_4, G, P, OVFL,	V_{OUT} = OV, V_{CC} = 5 V Only one Output at a time should be Shorted	20	50	90	20	50	90	mA
	Shift I/O	V_{OUT} = OV, V_{CC} = 5 V Only one Output at a time should be Shorted	5	15	50	5	15	50	mA

D.C. CHARACTERISTICS (Cont'd)

PARAMETER		DEVICE PINS	CONDITIONS	5701			6701			UNITS
				MIN.	TYP.[1]	MAX.	MIN.	TYP.[1]	MAX	
V_{OH}	Output Voltage "High"	DO_1, DO_2, DO_3, DO_4, G, C_{N+4}	$I_O = -3.2$ mA, V_{CC} = Min High Stored	2.4			2.4			V
		Shift I/O, P, OVFL	$I_O = -500$ μA V_{CC} = Min High Stored	2.4	3.5		2.4	3.5		V
C_I	Input Capacitance	All Inputs	$V_{CC} = 5.0$ V, $V_I = 2.0$ V, $25°$C, 1 MHz		8			8		pF
C_O	Output Capacitance	All Outputs	$V_{CC} = 5.0$ V, $V_O = 2.0$ V, $25°$C, 1 MHz		8			8		pF

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

A. C. CHARACTERISTICS The maximum or minimum of all possible input and output conditions is specified with outputs sinking maximum current and driving a 30pF load (15pF on Shift I/O).

PARAMETER	SYMBOL/CONDITIONS	5 V ± 10% −55 TO 125°C 5701			5 V ± 5% 0 TO 75°C 6701			UNITS
	SEE FIGURE 5	MIN.	TYP.[1]	MAX.	MIN.	TYP.[1]	MAX.	
Microinstruction Clock Cycle (Time Between Instructions)	T_{CYCLE}	250	152	∞	200	152	∞	ns
Accumulator Address to Data Out Delay Thru ALU/Bypass ALU	T_{AA}	40/20	110/60	170/90	40/20	110/60	140/85	ns
Instruction to Data Out Delay	T_{ID}	40	110	175	40	110	140	ns
Clock Pulse Width	T_{CW}	80	42		60	42		ns
Clock Set-up Time	T_{CS}	115	80		80	70		ns
3 State Enable to Data Out Enable	T_{EA}	10	25	40	10	25	37	ns
3 State Disable to Output Disable	T_{ER}	5	14	25	5	14	20	ns
C_N to C_{N+4}			20	30		20	30	ns
Accumulator Address to C_{N+4}	Clock High		65	95		65	90	ns
Data In to C_{N+4}			28	42		28	40	ns
Instruction to C_{N+4}	Clock High		54	81		54	78	ns
Accumulator Address to G	Clock High		60	90		60	85	ns
Data In to G			36	54		36	50	ns
Instruction to G or P	Clock High		52	78		52	75	ns
Accumulator Address to P	Clock High		60	90		60	85	ns
Data In to P			25	40		25	37	ns
Accumulator Address to F = All Highs	Clock High		64	96		64	90	ns
Data In to F = All Highs			60	90		60	86	ns
Instruction to F = All Highs	Clock High		76	110		76	100	ns
Accumulator Address to F = All Lows	Clock High		64	96		64	90	ns
Data In to F = All Lows			40	60		40	57	ns
Instruction to F = All Lows	Clock High		60	90		60	85	ns
Accumulator Address to SLO/SRI or SRO/SLI	Clock High		95	140		95	130	ns
Data In to SLO/SRI or SRO/SLI			40	60		40	55	ns
Instruction to SLO/SRI or SRO/SLI	Clock High		75	110		75	105	ns
Instruction to SQRI/SQLO or SQRO/SQLI	Clock High		25	40		25	37	ns
Clock to Data Out Delay	A_N, B_N, C_N, D_N Stable	40	110	175	40	110	140	ns

A.C. CHARACTERISTICS (Cont'd)

PARAMETER	SYMBOL/CONDITIONS SEE FIGURE 5	5 V ± 10% −55 TO 125°C 5701			5 V ± 5% 0 TO 75°C 6701			UNITS
		MIN.	TYP.[1]	MAX.	MIN.	TYP.[1]	MAX.	
Data In to Data Out Delay	T_{DD} Clock High	20	60	90	20	60	85	ns
Clock to Data In Hold Time	T_{CD} Clock Low to High Transition		−20	−10		−20	−10	ns
Latest Possible Arrival of Data In	T_{DI} Measure Relative to A_N or I_N			60			60	ns
Q Register to SQLO/SQRI or SQRO/SQRI	Measured From Low to High Transition of Clock		36	54		36	50	ns
Accumulator Address to Overflow	Clock High		90	135		90	125	ns
Data In to Overflow			36	54		36	50	ns
Instruction to Overflow	Clock High		56	84		56	80	ns
C_N to Overflow			25	38		25	35	ns
C_N to Data Out	Three State Low	5	36	54	5	36	50	ns

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

TIMING DIAGRAM
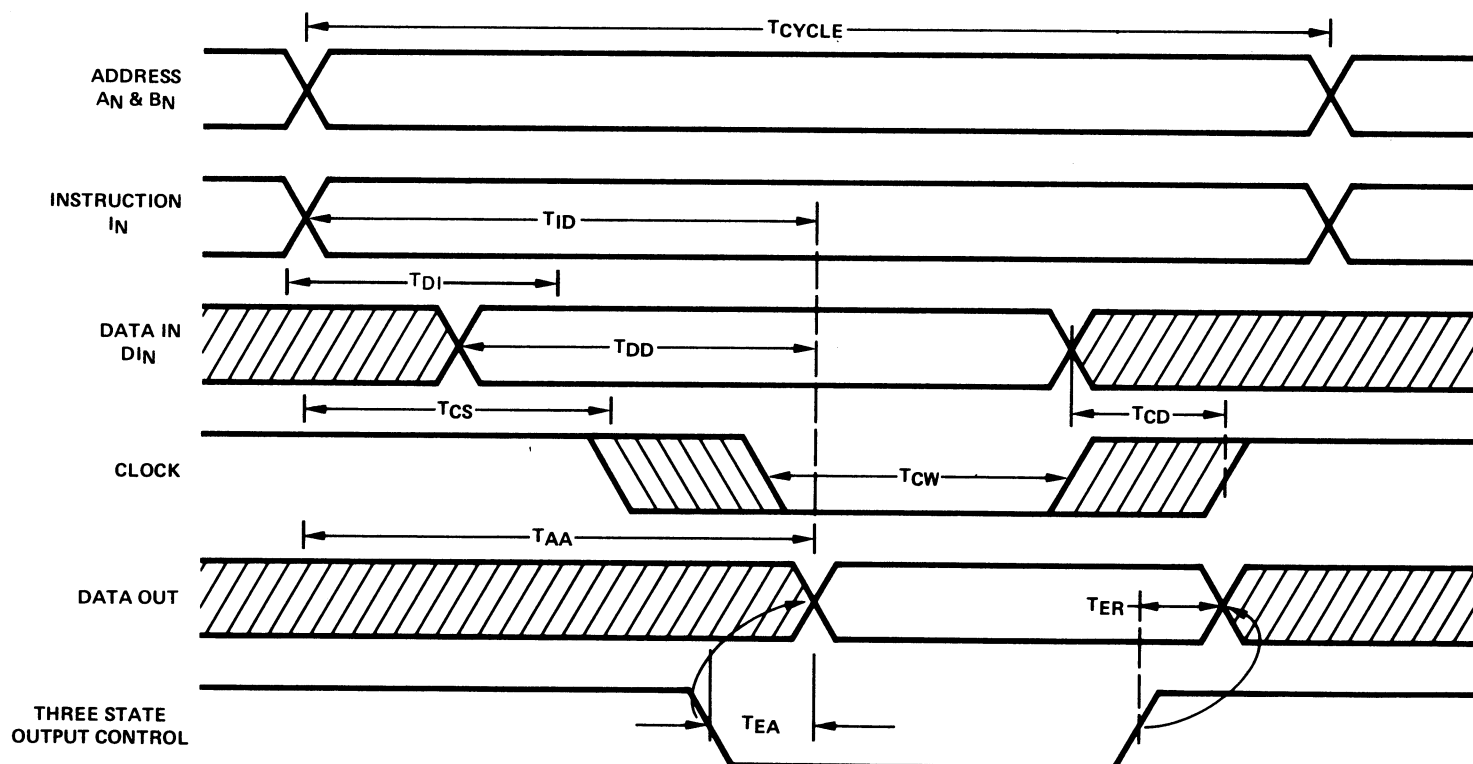
SAMPLE MINIMUM CYCLE TIMING DIAGRAM FOR 6701 (4 BIT MACHINE)
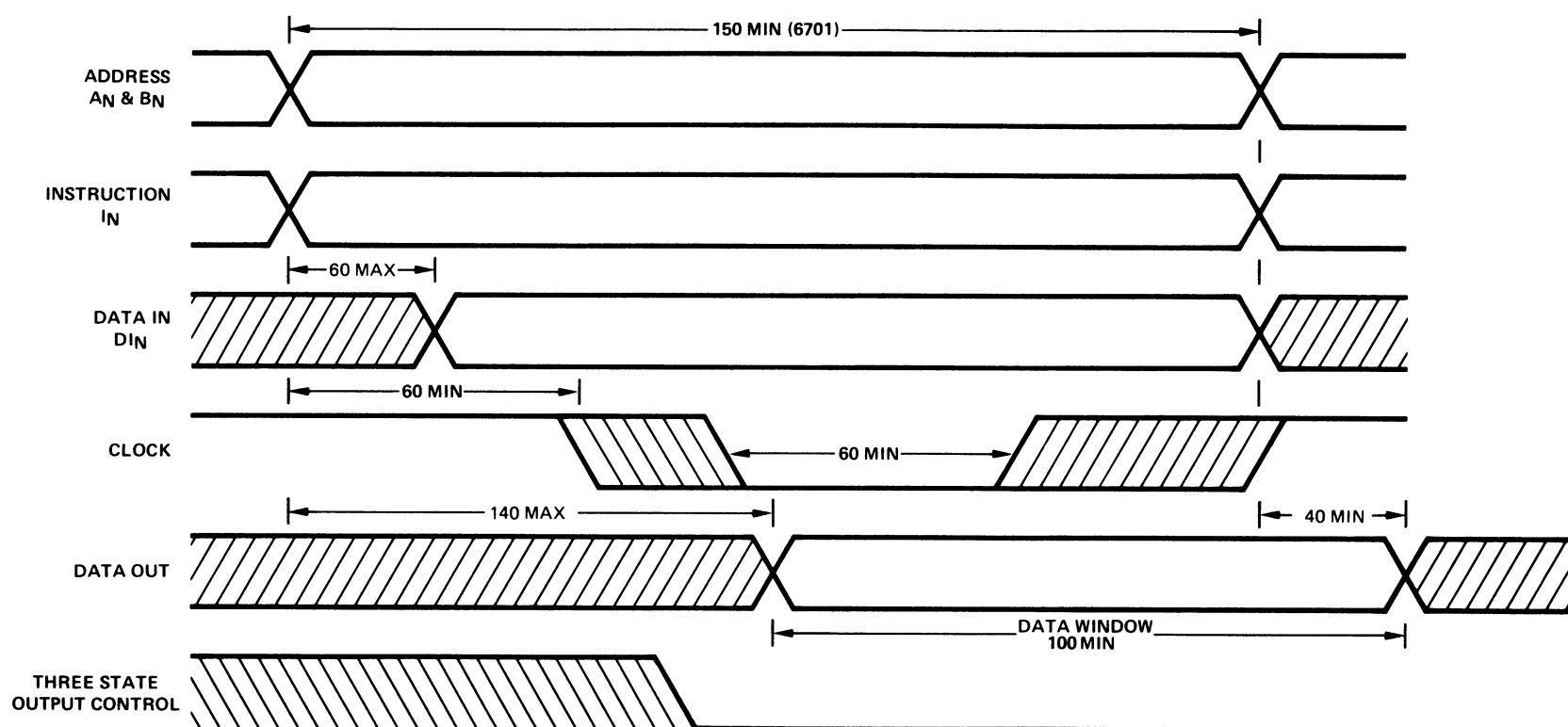
LEGEND: Shaded areas indicate don't care conditions or permitted timing tolerances.

Indicates that one pulse edge causes the other and that the two edges track.

FIGURE 5

INSTRUCTIONS LOCATED IN THE ON CHIP ROMS

SYMBOL DEFINITIONS

A_I = Any of the 16 four bit registers in the multiport RAM (I = O to 15)

B_J =   "   "   (J = 0 to 15)

$A_I + B_J$ = A_I plus B_J (arithmetic addition)

$A_I \veebar B_J$ = 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

$A_I \longrightarrow Q$ = Transfer A_I to Q, A_I saved, Old Q is lost

$A_I \longrightarrow OUT$ = Transfer A_I to the output pins, A_I is saved

$\overrightarrow{A_I}$ = A_I shifted right one bit

$\overleftarrow{A_I}$ = 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 "I" and a Low TTL Level ($\approx$ GRD) to be a "O". Negative logic defines a TTL High to be A "O" and a TTL Low to be a "I". Consider a 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
O	O	O
O	O	I
O	I	O
I	I	I

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

OUTPUT	INPUTS	
	B	A
I	I	I
I	I	O
I	O	I
O	O	O

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

POSITIVE LOGIC	NEGATIVE LOGIC
AND	OR
OR	AND
EXCLUSIVE OR	EXCLUSIVE NOR
EXCLUSIVE NOR	EXCLUSIVE OR
TRANSFER	DECREMENT
DECREMENT	TRANSFER
INCREMENT	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.
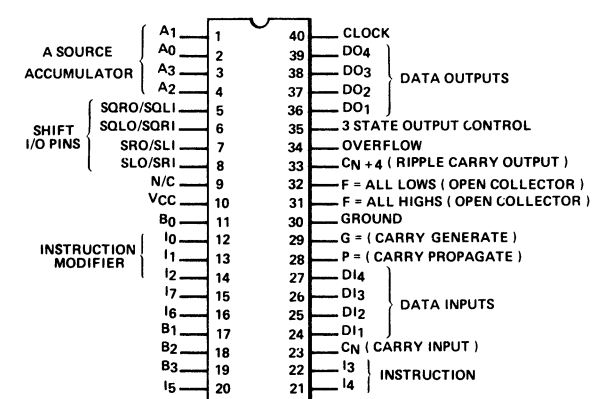
INSTRUCTIONS IN THE 32 x 9 ROM — POSITIVE LOGIC (1 = H ≈ 3 V) INTERPRETATION

I_7	I_6	I_5	I_4	I_3	Decimal	Octal	ALU Instruction (See Pg. 8 for Symbology)	ALU OUTPUT No Carry In (C_N = L)	ALU OUTPUT With Carry In (C_N = H)	TYPICAL USES	
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_I & B_j	$A_I \wedge B_j$	$A_I \wedge B_j$	AND	A_I & B_j
L	L	L	H	L	2	02	AND D_1 & B_j	$D_I \wedge B_j$	$D_I \wedge B_j$	''	D_I & B_j
L	L	L	H	H	3	03	OR A_I & B_j	$A_I \vee B_j$	$A_I \vee B_j$	OR	A_I & B_j
L	L	H	L	L	4	04	OR D_I & B_j	$D_I \vee B_j$	$D_I \vee B_j$	''	D_I & B_j
L	L	H	L	H	5	05	Exclusive OR A_I & B_j	$A_I \veebar B_j$	$A_I \veebar B_j$	Exclusive Or	A_I & B_j
L	L	H	H	L	6	06	Exclusive OR D_I & B_j	$D_I \veebar B_j$	$D_I \veebar B_j$	''	D_I & B_j
L	L	H	H	H	7	07	$\overline{A_I}$ + HHHH + C_N	$\overline{A_I}$ + 1111	$\overline{A_I}$	Invert	A_I
L	H	L	L	L	8	10	$\overline{D_I}$ + HHHH + C_N	$\overline{D_I}$ + 1111	$\overline{D_I}$	''	D_I
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	$\overline{A_I}$ + LLLL + C_N	$\overline{A_I}$	$\overline{A_I}$ + 0001	2's Complement Of	A_I
L	H	H	L	L	12	14	$\overline{D_I}$ + LLLL + C_N	$\overline{D_I}$	$\overline{D_I}$ + 0001	''	D_I
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_I + LLLL + C_N	A_I	A_I + 0001	Transfer Or Increment	A_I
H	L	L	L	L	16	20	D_I + LLLL + C_N	D_I	D_I + 0001	''	D_I
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_I + HHHH + C_N	A_I + 1111	A_I	Decrement Or Transfer	A_I
H	L	H	L	L	20	24	D_I + HHHH + C_N	D_I + 1111	D_I	''	D_I
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_I + B_j + C_N	A_I + B_j	A_I + B_j + 0001	Add	A_I & B_j
H	H	L	L	L	24	30	D_I + B_j + C_N	D_I + B_j	D_I + Q + 0001	''	D_I & Q
H	H	L	L	H	25	31	A_I + Q + C_N	A_I + Q	D_I + Q + 0001	''	A_I & Q
H	H	L	H	L	26	32	D_I + Q + C_N	D_I + Q	D_I + Q + 0001	''	D_I & Q
H	H	L	H	H	27	33	A_I + $\overline{B_j}$ + C_N	A_I − B_j − 0001	A_I − B_j	Subtract	A_I & B_j
H	H	H	L	L	28	34	B_j + $\overline{A_I}$ + C_N	B_j − A_I − 0001	B_j − A_I	''	B_j & A_I
H	H	H	L	H	29	35	D_I + $\overline{B_j}$ + C_N	D_I − B_j − 0001	D_I − B_j	''	D_I & B_j
H	H	H	H	L	30	36	B_j + $\overline{D_I}$ + C_N	B_j − D_I − 0001	B_j − D_I	''	B_j & D_I
H	H	H	H	H	31	37	D_I + $\overline{Q}$ + C_N	D_I − Q − 0001	D_I − Q	''	D_I & Q

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

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

PIN CONFIGURATION

Left pins		Right pins
A SOURCE ACCUMULATOR { A1 — 1		40 — CLOCK
A0 — 2		39 — DO4
A3 — 3		38 — DO3 — DATA OUTPUTS
A2 — 4		37 — DO2
SQRO/SQLI — 5		36 — DO1
SHIFT I/O PINS { SQLO/SQRI — 6		35 — 3 STATE OUTPUT CONTROL
SRO/SLI — 7		34 — OVERFLOW
SLO/SRI — 8		33 — C_N +4 (RIPPLE CARRY OUTPUT)
N/C — 9		32 — F = ALL LOWS (OPEN COLLECTOR)
VCC — 10		31 — F = ALL HIGHS (OPEN COLLECTOR)
B0 — 11		30 — GROUND
I0 — 12		29 — G = (CARRY GENERATE)
INSTRUCTION MODIFIER { I1 — 13		28 — P = (CARRY PROPAGATE)
I2 — 14		27 — DI4
I7 — 15		26 — DI3 — DATA INPUTS
I6 — 16		25 — DI2
B1 — 17		24 — DI1
B2 — 18		23 — C_N (CARRY INPUT)
B3 — 19		22 — I3 — INSTRUCTION
I5 — 20		21 — I4

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

I_7	I_6	I_5	I_4	I_3	Decimal	Octal	ALU Instruction (See Pg. 8 for Symbology)	ALU OUTPUT No Carry In (C_N = H)	ALU OUTPUT With Carry In (C_N = L)	TYPICAL USES	
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	37	OR A_I & B_j	$A_I \lor B_j$	$A_I \lor B_j$	OR	A_I & B_j
L	L	L	H	L	29	35	OR D_1 & B_j	$D_I \lor B_j$	$D_I \lor B_j$	''	D_I & B_j
L	L	L	H	H	28	34	AND A_I & B_j	$A_I \land B_j$	$A_I \land B_j$	AND	A_I & B_j
L	L	H	L	L	27	33	AND D_I & B_j	$D_I \land B_j$	$D_I \land B_j$	''	D_I & B_j
L	L	H	L	H	26	32	Exclusive OR A_I & B_j	$A_I \veebar B_j$	$A_I \veebar B_j$	Exclusive Nor	A_I & B_j
L	L	H	H	L	25	31	Exclusive OR D_I & B_j	$D_I \veebar B_j$	$D_I \veebar B_j$	''	D_I & B_j
L	L	H	H	H	24	30	$\overline{A_I}$ + HHHH + C_N	$\overline{A_I}$	$\overline{A_I}$ + 0001	2's Complement Of	A_I
L	H	L	L	L	23	27	$\overline{D_I}$ + HHHH + C_N	$\overline{D_I}$	$\overline{D_I}$ + 0001	''	D_I
L	H	L	L	H	22	26	$\overline{B_j}$ + HHHH + C_N	$\overline{B_j}$	$\overline{B_j}$ + 0001	''	B_j
L	H	L	H	L	21	25	$\overline{Q}$ + HHHH + C_N	$\overline{Q}$	$\overline{Q}$ + 0001	''	Q
L	H	L	H	H	20	24	$\overline{A_I}$ + LLLL + C_N	$\overline{A_I}$ + 1111	$\overline{A_I}$	Invert	A_I
L	H	H	L	L	19	23	$\overline{D_I}$ + LLLL + C_N	$\overline{D_I}$ + 1111	$\overline{D_I}$	''	D_I
L	H	H	L	H	18	22	$\overline{B_j}$ + LLLL + C_N	$\overline{B_j}$ + 1111	$\overline{B_j}$	''	B_j
L	H	H	H	L	17	21	$\overline{Q}$ + LLLL + C_N	$\overline{Q}$ + 1111	$\overline{Q}$	''	Q
L	H	H	H	H	16	20	A_I + LLLL + C_N	A_I + 1111	A_I	Decrement Or Transfer	A_I
H	L	L	L	L	15	17	D_I + LLLL + C_N	D_I + 1111	D_I	''	D_I
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_I + HHHH + C_N	A_I	A_I + 0001	Transfer Or Increment	A_I
H	L	H	L	L	11	13	D_I + HHHH + C_N	D_I	D_I + 0001	''	D_I
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_I + B_j + C_N	A_I + B_j	A_I + B_j + 0001	Add	A_I & B_j
H	H	L	L	L	7	07	D_I + B_j + C_N	D_I + B_j	D_I + Q + 0001	''	D_I & Q
H	H	L	L	H	6	06	A_I + Q + C_N	A_I + Q	D_I + Q + 0001	''	A_I & Q
H	H	L	H	L	5	05	D_I + Q + C_N	D_I + Q	D_I + Q + 0001	''	D_I & Q
H	H	L	H	H	4	04	A_I + B_j + C_N	A_I − B_j − 0001	A_I − B_j	Subtract	A_I & B_j
H	H	H	L	L	3	03	B_j + A_I + C_N	B_j − A_I − 0001	B_j − A_I	''	B_j & A_I
H	H	H	L	H	2	02	D_I + B_j + C_N	D_I − B_j − 0001	D_I − B_j	''	D_I & B_j
H	H	H	H	L	1	01	B_j + D_I + C_N	B_j − D_I − 0001	B_j − D_I	''	B_j & D_I
H	H	H	H	H	0	00	D_I + Q + C_N	D_I − Q − 0001	D_I − Q	''	D_I & Q

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

I_2	I_1	I_0	Rom Word Decimal	Load Control Load Ram B_j	Load Control Load Q	Shift Control Shift Left	Shift Control Shift Right	Shift Control Don't Shift	Data Out Control A Latch	Data Out Control B Latch	Data Out Control ALU Output F
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

PIN CONFIGURATION

Left pins	No.	No.	Right pins
A SOURCE ACCUMULATOR	A1 — 1	40 — CLOCK	
	A0 — 2	39 — DO4	DATA OUTPUTS
	A3 — 3	38 — DO3	
	A2 — 4	37 — DO2	
SHIFT I/O PINS	SQRO/SQLI — 5	36 — DO1	
	SQLO/SQRI — 6	35 — 3 STATE OUTPUT CONTROL	
	SRO/SLI — 7	34 — OVERFLOW	
	SLO/SRI — 8	33 — C_{N+4} (RIPPLE CARRY OUTPUT)	
	N/C — 9	32 — F = ALL HIGHS (RIPPLE CARRY OUTPUT)	
	Vcc — 10	31 — F = ALL LOWS (OPEN COLLECTOR)	
	B0 — 11	30 — GROUND	
INSTRUCTION MODIFIER	I0 — 12	29 — G = (CARRY GENERATE)	
	I1 — 13	28 — P = (CARRY PROPAGATE)	
	I2 — 14	27 — DI4	
	I7 — 15	26 — DI3	
	I6 — 16	25 — DI2	DATA INPUTS
	B1 — 17	24 — DI1	
	B2 — 18	23 — C_N (CARRY INPUT)	
	B3 — 19	22 — I3	
	I5 — 20	21 — I4	INSTRUCTION

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) $A_I \not\vee B_J$ 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,

   $B_O + OOOI \rightarrow B_O$ (Increment P.C)

   $A_I \rightarrow$ OUT (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 micro-controller cycle.

CARRY GENERATE (G) AND CARRY PROPAGATE (P)

A) THEORY

The G and 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

G and 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).
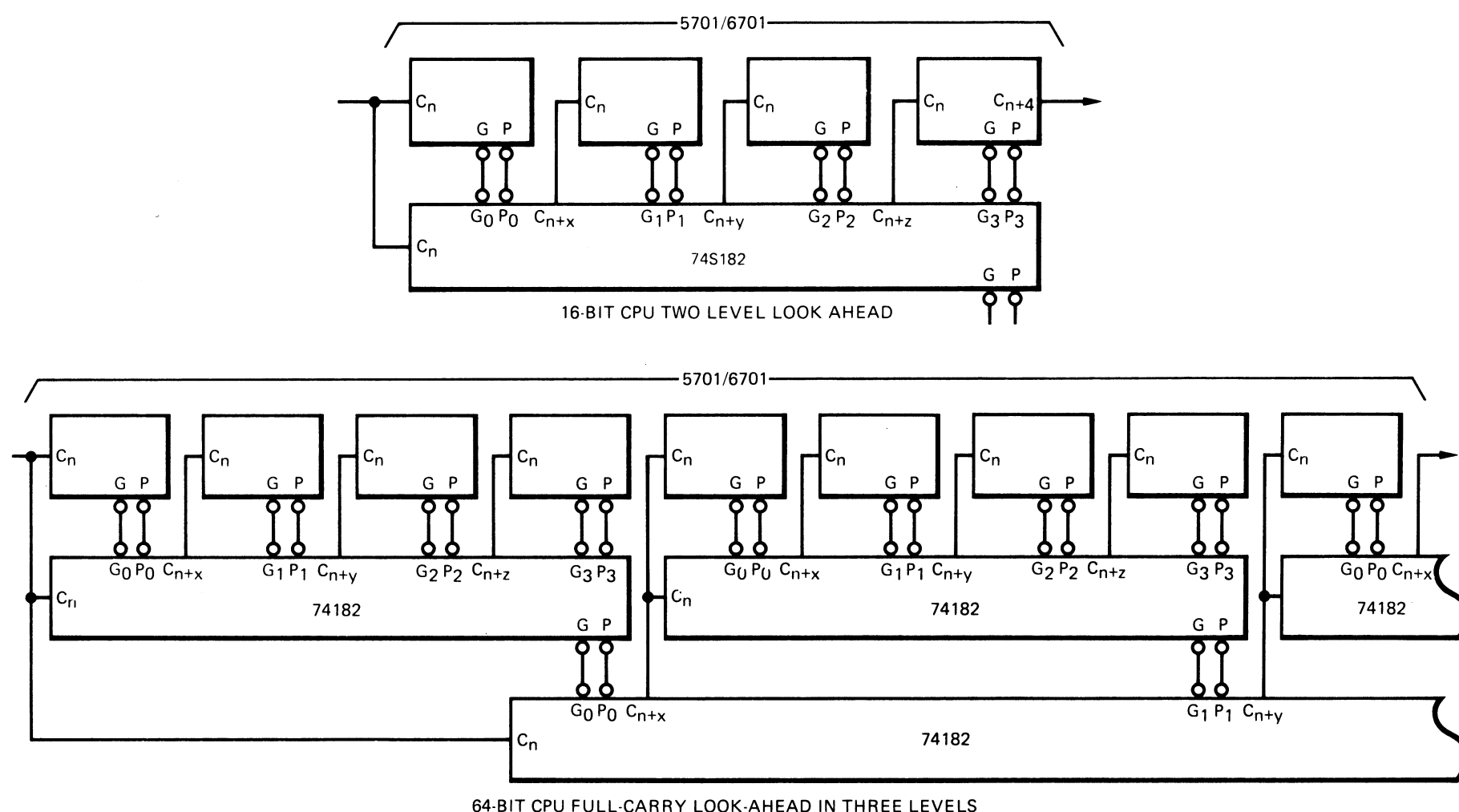

16-BIT CPU TWO LEVEL LOOK AHEAD

64-BIT CPU FULL-CARRY LOOK-AHEAD IN THREE LEVELS

FIGURE 6

12

C) LOGIC EQUATION

In logic symbology where J 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 microcontroller package to the carry input C_N of the more significant microcontroller 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.
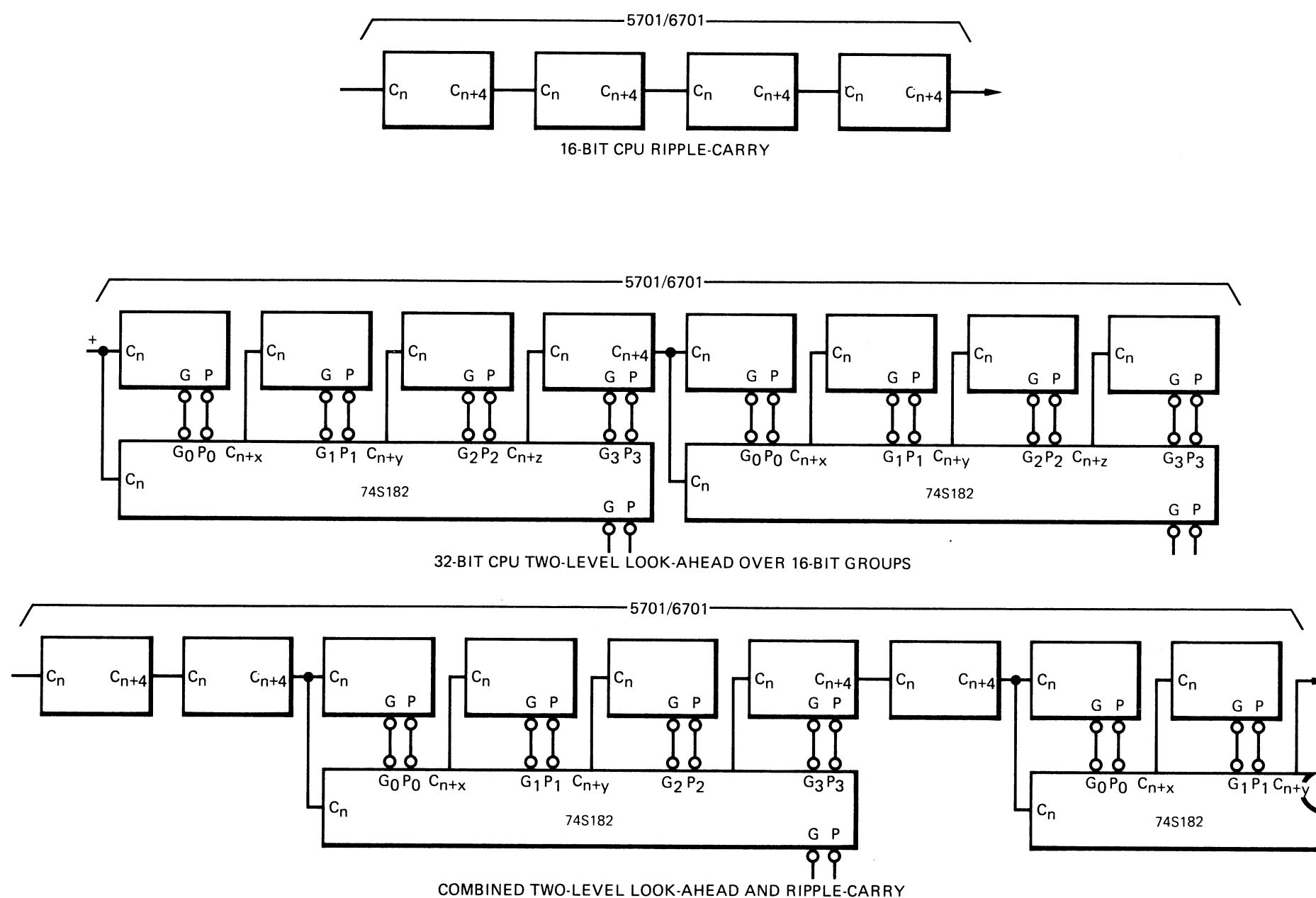

16-BIT CPU RIPPLE-CARRY

32-BIT CPU TWO-LEVEL LOOK-AHEAD OVER 16-BIT GROUPS

COMBINED TWO-LEVEL LOOK-AHEAD AND RIPPLE-CARRY

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 74S182's	Cycle Time (ns)		#Look Ahead 74S182's	Cycle Time (ns)
4	200			250		
8	200			250		
		1	205		1	250
12	230			259		
		1	205		1	250
16	260			289		
		1	205		1	250
24	320			349		
		1	245		1	284
		2	215		2	254
32	380			409		
		1	300		1	334
		2	245		2	284
		3	215		3	254
48	500			529		
		1	420		1	454
		2	340		2	379
		3	285		3	329
		4	215		4	254
64	620			649		
		1	540		1	574
		2	460		2	499
		3	380		3	424
		4	325		4	374
		5	215		5	254

*The minimum look-ahead cycle times assume a 74S182 with a delay of 10ns maximum for 5 V ± 5%, 0° to 75°C or a 54S182 with a delay of 15ns maximum for 5 V ±10%, –55° to 125°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_I or B_J to data output can be assumed to equal the delay from A_I or B_J to the data inputs of the on chip RAM. The increase in the delay from A_I 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 13).

From page 7 the accumulator address to G or P delay is 85ns it will then take another 10ns in the 74S182 to generate C_{N+X}, C_{N+Y}, and C_{N+Z} the C_N to output delay (50ns) is faster than the accumulator to data out delay (85ns) in the least significant package hence the 85ns path will limit the speed of the 74S182.

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 hook up 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.
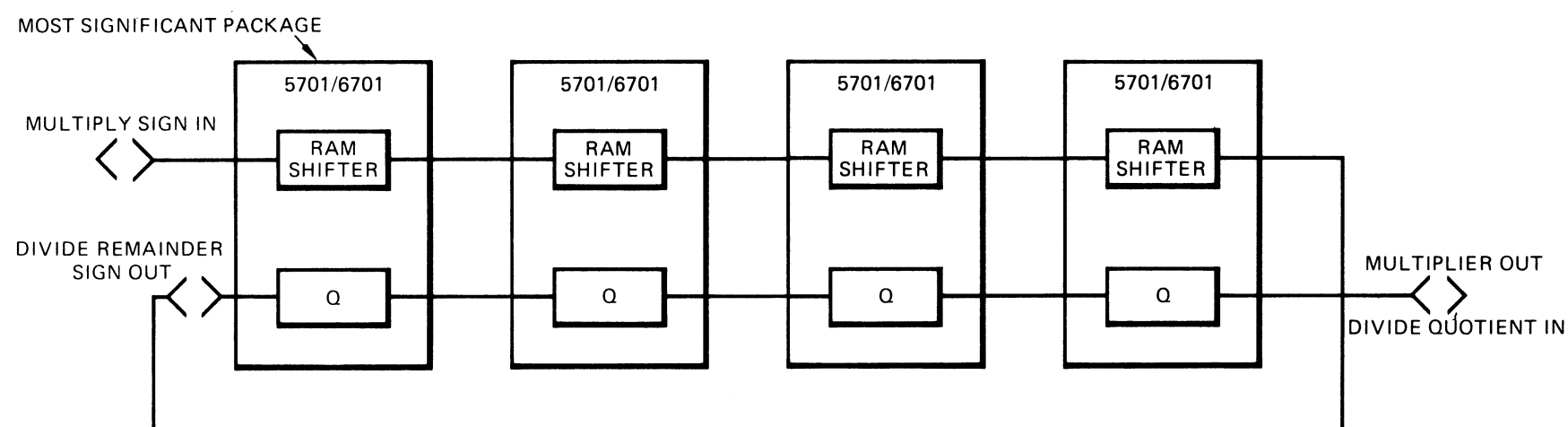

FIGURE 8

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 OR of the most significant bit of B and the most significant bit of the ALU output. After 16 reduction cycles, an additional correction cycle is required if the multipliers 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.

With C_{N+Z} generated at 95ns, the C_N to C_{N+4} delay (30ns) of package 4 (bits 12 to 16) is in the critical path. Generation of C_N to package 5 will arrive at 125ns. C_{N+Z} into package 8 will arrive at 135ns. The C_N to output delay of package 8 (50ns) will now let the data out of the package reflect the correct outputs by 185ns. Since the F bus of the 6701 ALU has gone thru the output buffers by 185ns, we can assume that by 185ns the F bus has been shifted and is stable at the data inputs of the 6701's RAM. We must now bring the clock low for 60ns from time 185 to 245ns to write the shifted result in the RAM. The complete add, shift, and store cycle is hence 245ns. The data outputs of package 8 (bits 28 to 32) will reflect the results of the addition in 185ns. Note that the chart above reflects a 245ns cycle time for two 74S182's and a 32 bit CPU.

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:

Overflow ◄──────────────── │────────────────────────────────────│ ──────────► Overflow
                          –8   0   +7
                        1.000   0.000   0.111
                          ʌ   ʌ   ʌ

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 = –1, no overflow

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

   +7 minus –7 = +14, overflow occurs
   +2 minus –1 = +1, no overflow

OVERFLOW PIN ON THE MICROCONTROLLER

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 microcontroller package in systems larger than four bits. Since the overflow is implemented with an exclusive nor gate the microcontroller 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.

APPLICATION INFORMATION (Cont'd)

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 on 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 on 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 0000000001111111 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 multipler'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 package 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 instruction 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.
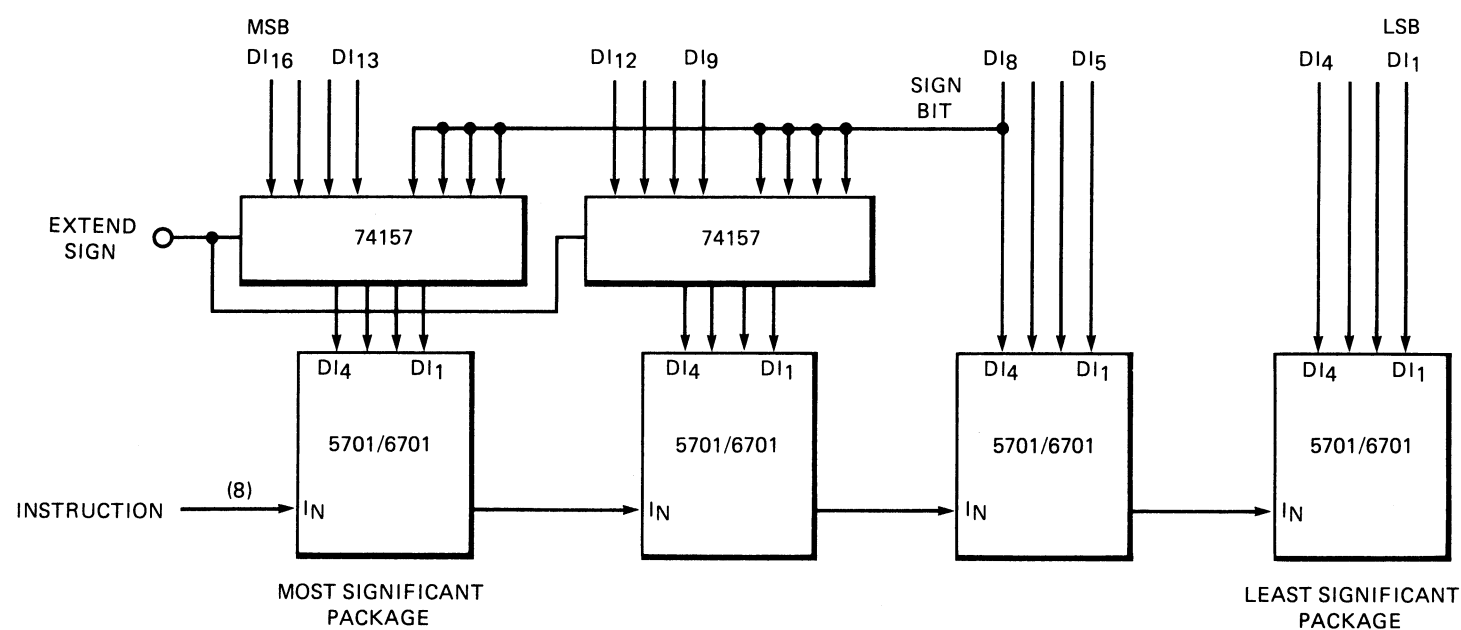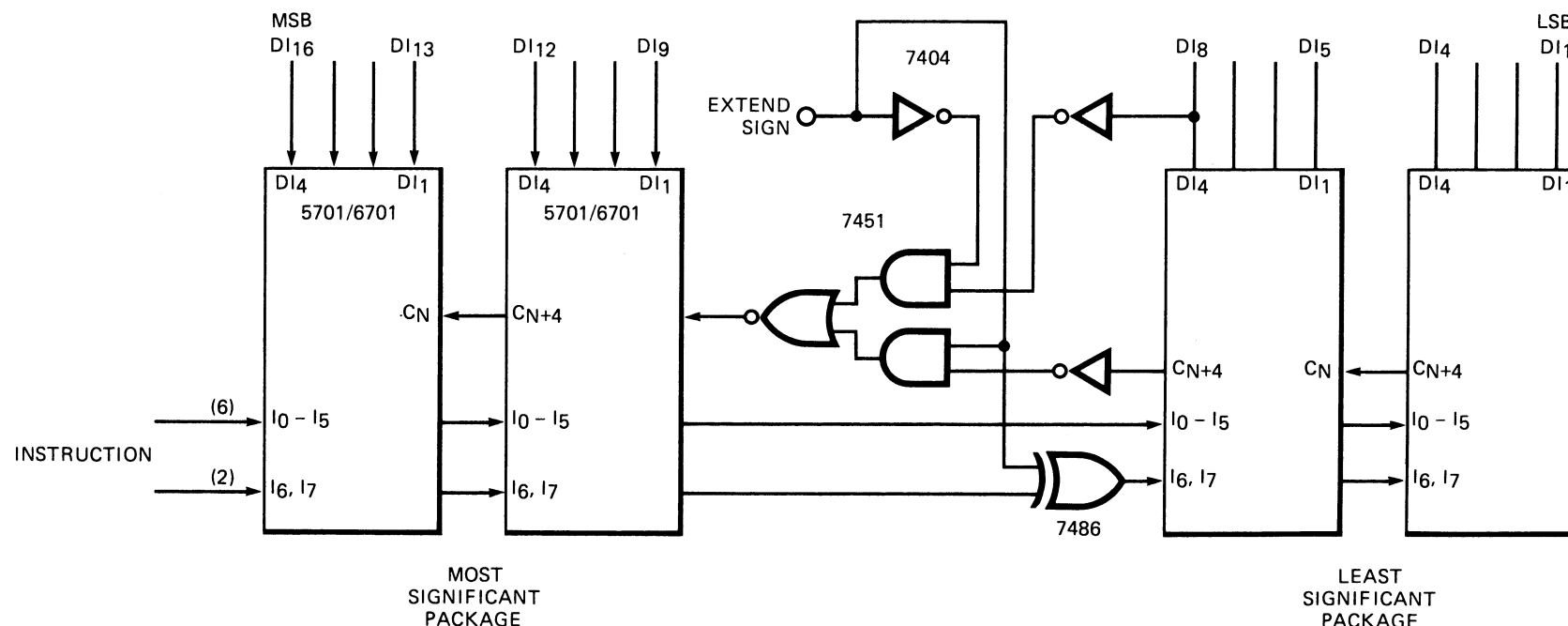

FIGURE 9

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.

Op Code	File	File
00011001	0101	0011

For example, an instruction register containing might mean add file 3 to file 5 and store the 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 micro-cycles 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.
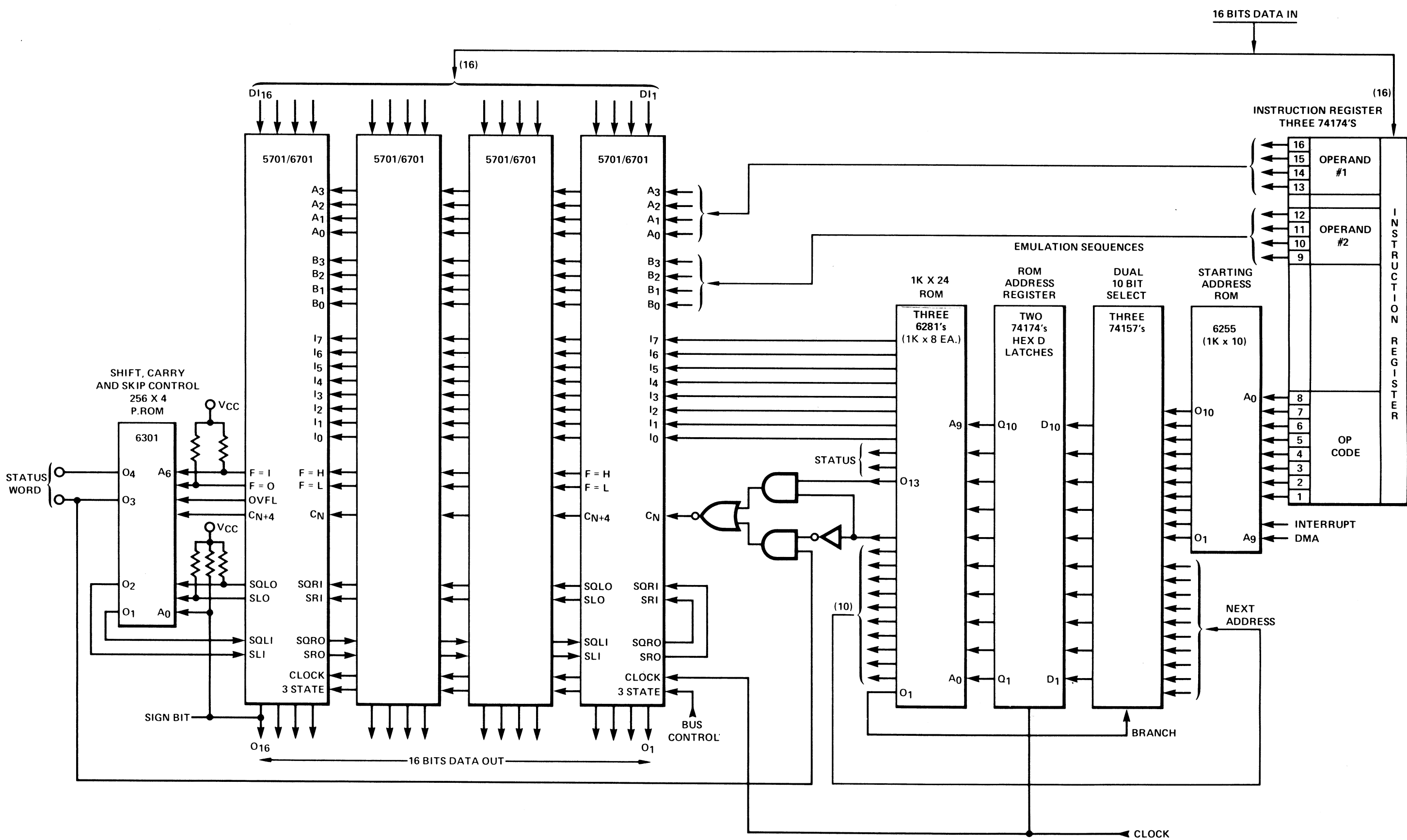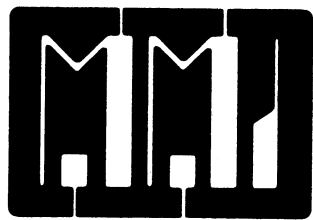
16 BIT MICROPROGRAMMED CPU

FIGURE 11

MMI

Monolithic Memories
INCORPORATED

Thank you for your recent inquiry on our NEW BIPOLAR 4-BIT
EXPANDABLE MICROCONTROLLER (5701/6701).

Enclosed you will find a complete specification sheet including
product features, applications, characteristics of the product,
etc. Pricing will be: 1-24; $195.00, 25-99; $150.00 and
100-999; $95.00.

Should you have further questions regarding our device, please
contact one of our local Area Managers (listed below) or our
facility for assistance.

Western Area	Marty Lyon
	(714) 556-1216
Central Area	Ron Rich
	(214) 369-7952
Eastern Area	Russell French
	(617) 653-3158

For latest information on Monolithic Memories, Inc. new
products please fill out and return the enclosed mailer. This
will enable you to receive data on our products as they are
announced.

Your interest in our company is greatly appreciated.

Very truly yours,

MONOLITHIC MEMORIES, INC.

D. W. Williams
Director of Marketing

Enclosures