intersil™

CDP1802A, CDP1802AC, CDP1802BC

March 1997

CMOS 8-Bit Microprocessors

Features

- **Maximum Input Clock Maximum Frequency Options At V_{DD} = 5V**
  - **CDP1802A, AC 3.2MHz**
  - **CDP1802BC 5.0MHz**
- **Maximum Input Clock Maximum Frequency Options At V_{DD} = 10V**
  - **CDP1802A, AC 6.4MHz**
- **Minimum Instruction Fetch-Execute Times At V_{DD} = 5V**
  - **CDP1802A, AC 5.0µs**
  - **CDP1802BC 3.2µs**
- **Any Combination of Standard RAM and ROM Up to 65,536 Bytes**
- **8-Bit Parallel Organization With Bidirectional Data Bus and Multiplexed Address Bus**
- **16 x 16 Matrix of Registers for Use as Multiple Program Counters, Data Pointers, or Data Registers**
- **On-Chip DMA, Interrupt, and Flag Inputs**
- **Programmable Single-Bit Output Port**
- **91 Easy-to-Use Instructions**

Description

The CDP1802 family of CMOS microprocessors are 8-bit register oriented central processing units (CPUs) designed for use as general purpose computing or control elements in a wide range of stored program systems or products.

The CDP1802 types include all of the circuits required for fetching, interpreting, and executing instructions which have been stored in standard types of memories. Extensive input/output (I/O) control features are also provided to facilitate system design.

The 1800 series architecture is designed with emphasis on the total microcomputer system as an integral entity so that systems having maximum flexibility and minimum cost can be realized. The 1800 series CPU also provides a synchronous interface to memories and external controllers for I/O devices, and minimizes the cost of interface controllers. Further, the I/O interface is capable of supporting devices operating in polled, interrupt driven, or direct memory access modes.

The CDP1802A and CDP1802AC have a maximum input clock frequency of 3.2MHz at V_{DD} = 5V. The CDP1802A and CDP1802AC are functionally identical. They differ in that the CDP1802A has a recommended operating voltage range of 4V to 10.5V, and the CDP1802AC a recommended operating voltage range of 4V to 6.5V.

The CDP1802BC is a higher speed version of the CDP1802AC, having a maximum input clock frequency of 5.0MHz at V_{DD} = 5V, and a recommended operating voltage range of 4V to 6.5V.

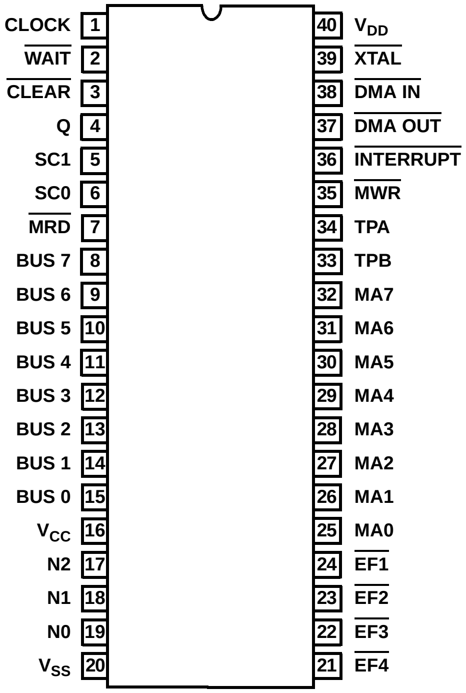
Ordering Information

PART NUMBER				
5V - 3.2MHz	**5V - 5MHz**	**TEMPERATURE RANGE**	**PACKAGE**	**PKG. NO.**
CDP1802ACE	CDP1802BCE	-40°C to +85°C	PDIP	E40.6
CDP1802ACEX	CDP1802BCEX		Burn-In	E40.6
CDP1802ACQ	CDP1802BCQ	-40°C to +85°C	PLCC	N44.65
CDP1802ACD	-	-40°C to +85°C	SBDIP	D40.6
CDP1802ACDX	CDP1802BCDX		Burn-In	D40.6

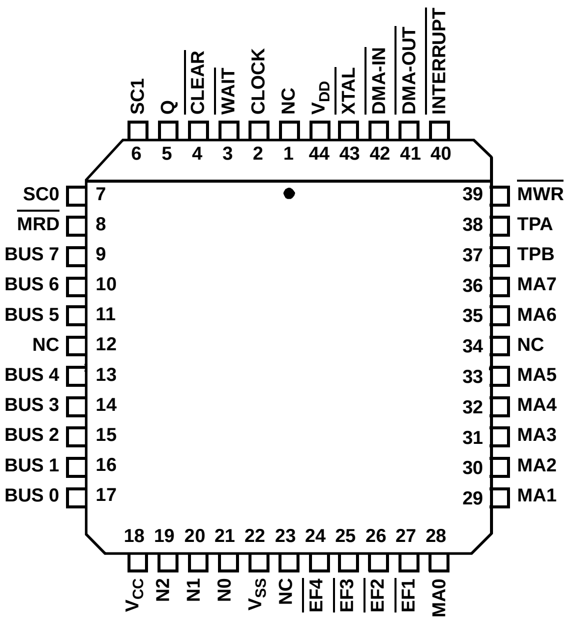
CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.
1-888-INTERSIL or 321-724-7143 | Intersil (and design) is a trademark of Intersil Americas Inc.

Pinouts

40 LEAD PDIP (PACKAGE SUFFIX E)
40 LEAD SBDIP (PACKAGE SUFFIX D)
TOP VIEW


CLOCK 1
$\overline{\text{WAIT}}$ 2
$\overline{\text{CLEAR}}$ 3
Q 4
SC1 5
SC0 6
$\overline{\text{MRD}}$ 7
BUS 7 8
BUS 6 9
BUS 5 10
BUS 4 11
BUS 3 12
BUS 2 13
BUS 1 14
BUS 0 15
V_{CC} 16
N2 17
N1 18
N0 19
V_{SS} 20
40 V_{DD}
39 $\overline{\text{XTAL}}$
38 $\overline{\text{DMA IN}}$
37 $\overline{\text{DMA OUT}}$
36 $\overline{\text{INTERRUPT}}$
35 $\overline{\text{MWR}}$
34 TPA
33 TPB
32 MA7
31 MA6
30 MA5
29 MA4
28 MA3
27 MA2
26 MA1
25 MA0
24 $\overline{\text{EF1}}$
23 $\overline{\text{EF2}}$
22 $\overline{\text{EF3}}$
21 $\overline{\text{EF4}}$

44 LEAD PLCC
(PACKAGE TYPE Q)
TOP VIEW


SC1 6
Q 5
$\overline{\text{CLEAR}}$ 4
$\overline{\text{WAIT}}$ 3
CLOCK 2
NC 1
V_{DD} 44
$\overline{\text{XTAL}}$ 43
$\overline{\text{DMA-IN}}$ 42
$\overline{\text{DMA-OUT}}$ 41
$\overline{\text{INTERRUPT}}$ 40
SC0 7
$\overline{\text{MRD}}$ 8
BUS 7 9
BUS 6 10
BUS 5 11
NC 12
BUS 4 13
BUS 3 14
BUS 2 15
BUS 1 16
BUS 0 17
39 $\overline{\text{MWR}}$
38 TPA
37 TPB
36 MA7
35 MA6
34 NC
33 MA5
32 MA4
31 MA3
30 MA2
29 MA1
18 V_{CC}
19 N2
20 N1
21 N0
22 V_{SS}
23 NC
24 $\overline{\text{EF4}}$
25 $\overline{\text{EF3}}$
26 $\overline{\text{EF2}}$
27 $\overline{\text{EF1}}$
28 MA0

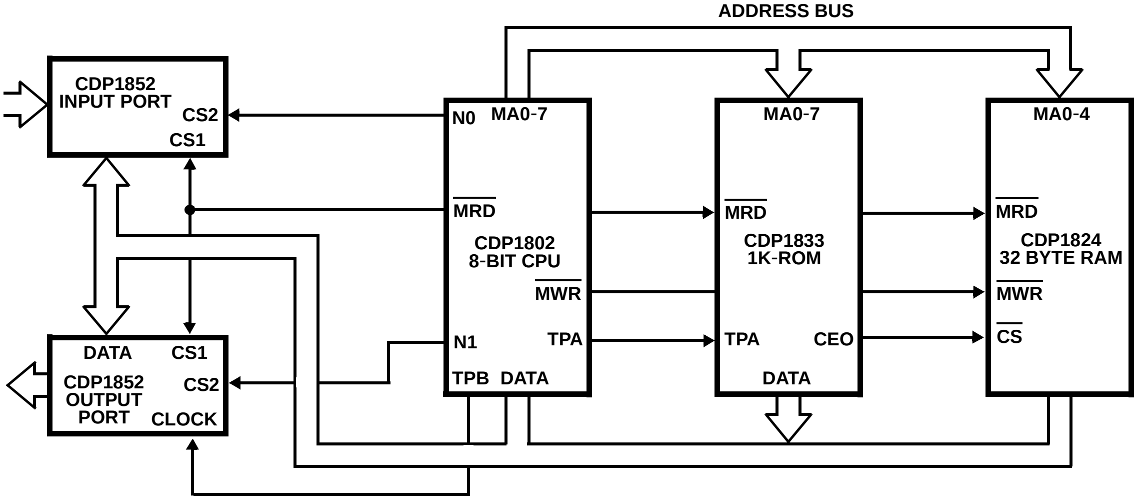


FIGURE 1. TYPICAL CDP1802 SMALL MICROPROCESSOR SYSTEM

Block Diagram

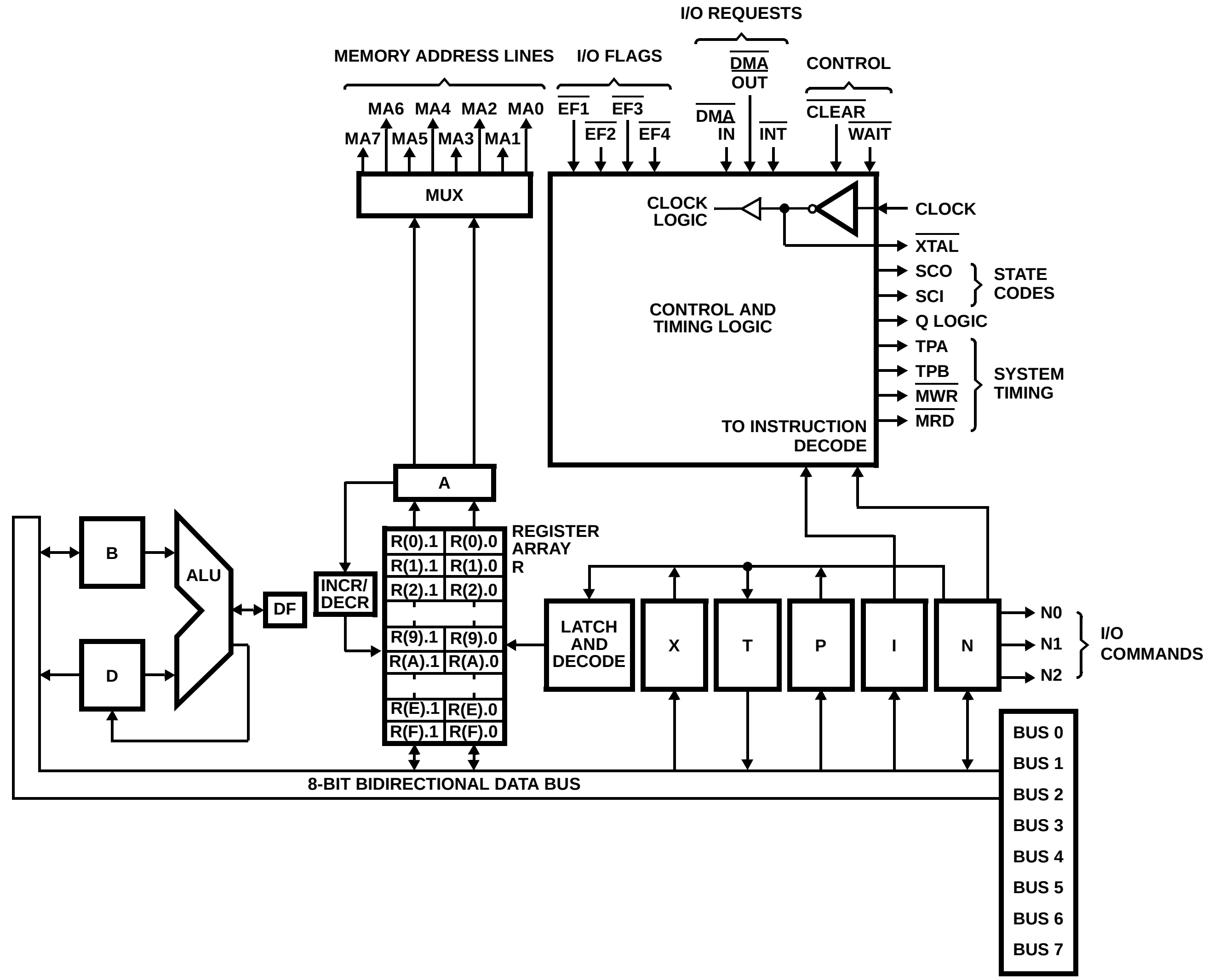


FIGURE 2.

Absolute Maximum Ratings

DC Supply Voltage Range, (V_{DD})
(All Voltages Referenced to V_{SS} Terminal)
CDP1802A. . . . . -0.5V to +11V
CDP1802AC, CDP1802BC. -0.5V to +7V
Input Voltage Range, All Inputs . . . . -0.5V to V_{DD} +0.5V
DC Input Current, any One Input. . . . . ±10mA

Thermal Information

Thermal Resistance (Typical, Note 4)	θ_{JA} (^oC/W)	θ_{JC} (^oC/W)
PDIP	50	N/A
PLCC.	46	N/A
SBDIP	55	15

Device Dissipation Per Output Transistor
T_A = Full Package Temperature Range. 100mW
Operating Temperature Range (T_A)
Package Type D . . . . -55^{o}C to +125^{o}C
Package Type E and Q. . . . . -40^{o}C to +85^{o}C
Storage Temperature Range (T_{STG}) . . . . -65^{o}C to +150^{o}C
Lead Temperature (During Soldering)
At distance 1/16 ± 1/32 In. (1.59 ± 0.79mm)
from case for 10s max. . . . . +265^{o}C
Lead Tips Only. . . . . +300^{o}C

CAUTION: Stresses above those listed in “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Recommended Operating Conditions

T_A = -40^{o}C to +85^{o}C. For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

PARAMETER	TEST CONDITIONS		CDP1802A		CDP1802AC		CDP1802BC		UNITS
	(NOTE 2) **V_{CC} (V)**	**V_{DD} (V)**	MIN	MAX	MIN	MAX	MIN	MAX	
DC Operating Voltage Range	-	-	4	10.5	4	6.5	4	6.5	V
Input Voltage Range	-	-	V_{SS}	V_{DD}	V_{SS}	V_{DD}	V_{SS}	V_{DD}	V
Maximum Clock Input Rise or Fall Time	4 to 6.5	4 to 6.5	-	-	-	1	-	1	µs
	4 to 10.5	4 to 10.5	-	1	-	-	-	-	µs
Minimum Instruction Time (Note 3)	5	5	5	-	5	-	3.2	-	µs
	5	10	4	-	-	-	-	-	µs
	10	10	2.5	-	-	-	-	-	µs
Maximum DMA Transfer Rate	5	5	-	400	-	400	-	667	KBytes/s
	5	10	-	500	-	-	-	-	
	10	10	-	800	-	-	-	-	
Maximum Clock Input Frequency, f_{CL}, Load Capacitance (C_L) = 50pF	5	5	DC	3.2	DC	3.2	DC	5	MHz
	5	10	DC	4	-	-	-	-	MHz
	10	10	DC	6.4	-	-	-	-	MHz

NOTES:

1. Printed circuit board mount: 57mm x 57mm minimum area x 1.6mm thick G10 epoxy glass, or equivalent.
2. V_{CC} must never exceed V_{DD}.
3. Equals 2 machine cycles - one Fetch and one Execute operation for all instructions except Long Branch and Long Skip, which require 3 machine cycles - one Fetch and two Execute operations.
4. θ_{JA} is measured with component mounted on an evaluation board in free air.

Static Electrical Specifications at T_A = -40°C to +85°C, Except as Noted

		TEST CONDITIONS			CDP1802A			CDP1802AC, CDP1802BC			
PARAMETER	SYMBOL	V_{OUT} (V)	V_{IN} (V)	V_{CC}, V_{DD} (V)	MIN	(NOTE 1) TYP	MAX	MIN	(NOTE 1) TYP	MAX	UNITS
Quiescent Device Current	I_{DD}	-	-	5	-	0.1	50	-	1	200	µA
		-	-	10	-	1	200	-	-	-	µA
Output Low Drive (Sink) Current	I_{OL}	0.4	0, 5	5	1.1	2.2	-	1.1	2.2	-	mA
(Except $\overline{\text{XTAL}}$)		0.5	0, 10	10	2.2	4.4	-	-	-	-	mA
$\overline{\text{XTAL}}$		0.4	5	5	170	350	-	170	350	-	µA
Output High Drive (Source) Current	I_{OH}	4.6	0, 5	5	-0.27	-0.55	-	-0.27	-0.55	-	mA
(Except $\overline{\text{XTAL}}$)		9.5	0, 10	10	-0.55	-1.1	-	-	-	-	mA
$\overline{\text{XTAL}}$		4.6	0	5	-125	-250	-	-125	-250	-	µA
Output Voltage Low Level	V_{OL}	-	0, 5	5	-	0	0.1	-	0	0.1	V
		-	0, 10	10	-	0	0.1	-	-	-	V
Output Voltage High Level	V_{OH}	-	0, 5	5	4.9	5	-	4.9	5	-	V
		-	0, 10	10	9.9	10	-	-	-	-	V
Input Low Voltage	V_{IL}	0.5, 4.5	-	5	-	-	1.5	-	-	1.5	V
		0.5, 4.5	-	5, 10	-	-	1	-	-	-	V
		1, 9	-	10	-	-	3	-	-	-	V
Input High Voltage	V_{IH}	0.5, 4.5	-	5	3.5	-	-	3.5	-	-	V
		0.5, 4.5	-	5, 10	4	-	-	-	-	-	V
		1, 9	-	10	7	-	-	-	-	-	V
$\overline{\text{CLEAR}}$ Input Voltage	V_H	-	-	5	0.4	0.5	-	0.4	0.5	-	V
Schmitt Hysteresis		-	-	5, 10	0.3	0.4	-	-	-	-	V
		-	-	10	1.5	2	-	-	-	-	V
Input Leakage Current	I_{IN}	Any Input	0, 5	5	-	$\pm 10^{-4}$	±1	-	$\pm 10^{-4}$	±1	µA
			0, 10	10	-	$\pm 10^{-4}$	±1	-	-	-	µA
Three-State Output Leakage Current	I_{OUT}	0, 5	0, 5	5	-	$\pm 10^{-4}$	±1	-	$\pm 10^{-4}$	±1	µA
		0, 10	0, 10	10	-	$\pm 10^{-4}$	±1	-	-	-	µA
Operating Current CDP1802A, AC at f = 3.2MHz	I_{DDI} (Note 2)	-	-	5	-	2	4	-	2	4	mA
CDP1802BC at f = 5.0MHz		-	-	5	-	-	-	-	3	6	mA
Minimum Data Retention Voltage	V_{DR}	$V_{DD} = V_{DR}$			-	2	2.4	-	2	2.4	V
Data Retention Current	I_{DR}	V_{DD} = 2.4V			-	0.05	-	-	0.5	-	µA

Static Electrical Specifications at T_A = -40°C to +85°C, Except as Noted **(Continued)**

		TEST CONDITIONS			CDP1802A			CDP1802AC, CDP1802BC			
PARAMETER	SYMBOL	V_{OUT} (V)	V_{IN} (V)	V_{CC}, V_{DD} (V)	MIN	(NOTE 1) TYP	MAX	MIN	(NOTE 1) TYP	MAX	UNITS
Input Capacitance	C_{IN}				-	5	7.5	-	5	7.5	pF
Output Capacitance	C_{OUT}				-	10	15	-	10	15	pF

NOTES:

1. Typical values are for T_A = +25°C and nominal V_{DD}.
2. Idle "00" at M(0000), C_L = 50pF.

Dynamic Electrical Specifications T_A = -40°C to +85°C, C_L = 50pF, V_{DD} ±5%, Except as Noted

		TEST CONDITIONS		CDP1802A, CDP1802AC		CDP1802BC		
PARAMETER	SYMBOL	V_{CC} (V)	V_{DD} (V)	(NOTE 1) TYP	MAX	(NOTE 1) TYP	MAX	UNITS
PROPAGATION DELAY TIMES								
Clock to TPA, TPB	t_{PLH}, t_{PHL}	5	5	200	300	200	300	ns
		5	10	150	250	-	-	ns
		10	10	100	150	-	-	ns
Clock-to-Memory High-Address Byte	t_{PLH}, t_{PHL}	5	5	600	850	475	525	ns
		5	10	400	600	-	-	ns
		10	10	300	400	-	-	ns
Clock-to-Memory Low-Address Byte Valid	t_{PLH}, t_{PHL}	5	5	250	350	175	250	ns
		5	10	150	250	-	-	ns
		10	10	100	150	-	-	ns
Clock to $\overline{MRD}$	t_{PHL}	5	5	200	300	175	275	ns
		5	10	150	250	-	-	ns
		10	10	100	150	-	-	ns
Clock to $\overline{MRD}$	t_{PLH}	5	5	200	350	175	275	ns
		5	10	150	290	-	-	ns
		10	10	100	175	-	-	ns
Clock to $\overline{MWR}$	t_{PLH}, t_{PHL}	5	5	200	300	175	225	ns
		5	10	150	250	-	-	ns
		10	10	100	150	-	-	ns
Clock to (CPU DATA to BUS) Valid	t_{PLH}, t_{PHL}	5	5	300	450	250	375	ns
		5	10	250	350	-	-	ns
		10	10	100	200	-	-	ns

Dynamic Electrical Specifications T_A = -40°C to +85°C, C_L = 50pF, V_{DD} ±5%, Except as Noted **(Continued)**

		TEST CONDITIONS		CDP1802A, CDP1802AC		CDP1802BC		
PARAMETER	**SYMBOL**	**V_{CC} (V)**	**V_{DD} (V)**	(NOTE 1) **TYP**	**MAX**	(NOTE 1) **TYP**	**MAX**	**UNITS**
Clock to State Code	t_{PLH}, t_{PHL}	5	5	300	450	250	400	ns
		5	10	250	350	-	-	ns
		10	10	150	250	-	-	ns
Clock to Q	t_{PLH}, t_{PHL}	5	5	250	400	200	300	ns
		5	10	150	250	-	-	ns
		10	10	100	150	-	-	ns
Clock to N (0 - 2)	t_{PLH}, t_{PHL}	5	5	300	550	275	350	ns
		5	10	200	350	-	-	ns
		10	10	150	250	-	-	ns
MINIMUM SET UP AND HOLD TIMES								
Data Bus Input Set Up	t_{SU}	5	5	-20	25	-20	0	ns
		5	10	0	50	-	-	ns
		10	10	-10	40	-	-	ns
Data Bus Input Hold	t_H (Note 2)	5	5	150	200	125	150	ns
		5	10	100	125	-	-	ns
		10	10	75	100	-	-	ns
$\overline{\text{DMA}}$ Set Up	t_{SU}	5	5	0	30	0	30	ns
		5	10	0	20	-	-	ns
		10	10	0	10	-	-	ns
$\overline{\text{DMA}}$ Hold	t_H (Note 2)	5	5	150	250	100	150	ns
		5	10	100	200	-	-	ns
		10	10	75	125	-	-	ns
Interrupt Set Up	t_{SU}	5	5	-75	0	-75	0	ns
		5	10	-50	0	-	-	ns
		10	10	-25	0	-	-	ns
Interrupt Hold	t_H (Note 2)	5	5	100	150	75	125	ns
		5	10	75	100	-	-	ns
		10	10	50	75	-	-	ns
$\overline{\text{WAIT}}$ Set Up	t_{SU}	5	5	10	50	20	40	ns
		5	10	-10	15	-	-	ns
		10	10	0	25	-	-	ns

Dynamic Electrical Specifications T_A = -40°C to +85°C, C_L = 50pF, V_{DD} ±5%, Except as Noted **(Continued)**

		TEST CONDITIONS		CDP1802A, CDP1802AC		CDP1802BC		
PARAMETER	**SYMBOL**	V_{CC} (V)	V_{DD} (V)	(NOTE 1) TYP	MAX	(NOTE 1) TYP	MAX	UNITS
$\overline{\text{EF1-4}}$ Set Up	t_{SU}	5	5	-30	20	-30	0	ns
		5	10	-20	30	-	-	ns
		10	10	-10	40	-	-	ns
EF1-4 Hold	t_H (Note 2)	5	5	150	200	100	150	ns
		5	10	100	150	-	-	ns
		10	10	75	100	-	-	ns
Minimum Pulse Width Times								
$\overline{\text{CLEAR}}$ Pulse Width	t_{WL} (Note 2)	5	5	150	300	100	150	ns
		5	10	100	200	-	-	ns
		10	10	75	150	-	-	ns
$\overline{\text{CLOCK}}$ Pulse Width	t_{WL}	5	5	125	150	90	100	ns
		5	10	100	125	-	-	ns
		10	10	60	75	-	-	ns

NOTES:

1. Typical values are for T_A = +25°C and nominal V_{DD}.
2. Maximum limits of minimum characteristics are the values above which all devices function.

Timing Specifications as a function of T(T = 1/f_{CLOCK}) at T_A = -40 to +85°C, Except as Noted

		TEST CONDITIONS		CDP1802A, CDP1802AC		CDP1802BC		
PARAMETERS	**SYMBOL**	V_{CC} (V)	V_{DD} (V)	MIN	(NOTE 1) TYP	MIN	(NOTE 1) TYP	UNITS
High-Order Memory-Address Byte Set Up to TPA ↘ Time	t_{SU}	5	5	2T-550	2T-400	2T-325	2T-275	ns
		5	10	2T-350	2T250	-	-	ns
		10	10	2T-250	2T-200	-	-	ns
High-Order Memory-Address Byte Hold After TPA Time	t_H	5	5	t/2-25	T/2-15	T/2-25	T/2-15	ns
		5	10	T/2-35	T/2-25	-	-	ns
		10	10	T/2-10	T/2-+0	-	-	ns
Low-Order Memory-Address Byte Hold After WR Time	t_H	5	5	T-30	T+0	T-30	T+0	ns
		5	10	T-20	T+0	-	-	ns
		10	10	T-10	T+0	-	-	ns
CPU Data to Bus Hold After WR Time	t_H	5	5	T-200	T-150	T-175	T-125	ns
		5	10	T-150	T-100	-	-	ns
		10	10	T-100	T-50	-	-	ns

Timing Specifications as a function of T(T = 1/f_{CLOCK}) at T_A = -40 to +85^{o}C, Except as Noted

PARAMETERS	SYMBOL	TEST CONDITIONS		CDP1802A, CDP1802AC		CDP1802BC		UNITS
		V_{CC} (V)	V_{DD} (V)	MIN	(NOTE 1) TYP	MIN	(NOTE 1) TYP	
Required Memory Access Time Address to Data	t_{ACC}	5	5	5T-375	5T-250	5T-225	5T-175	ns
		5	10	5T-250	5T-150	-	-	ns
		10	10	5T-190	5T-100	-	-	ns
$\overline{MRD}$ to TPA ↘	t_{SU}	5	5	T/2-25	T/2-18	T/2-20	T/2-15	ns
		5	10	T/2-20	T/2-15	-	-	ns
		10	10	T/2-15	T/2-10	-	-	ns

NOTE:

1. Typical values are for T_A = +25^{o}C and nominal V_{DD}.

Timing Waveforms

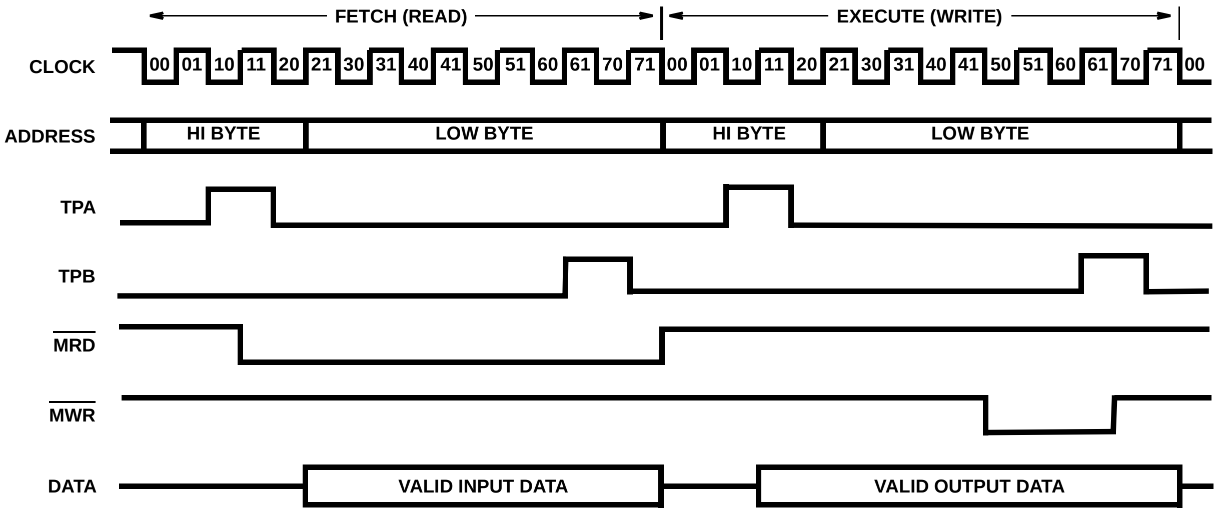
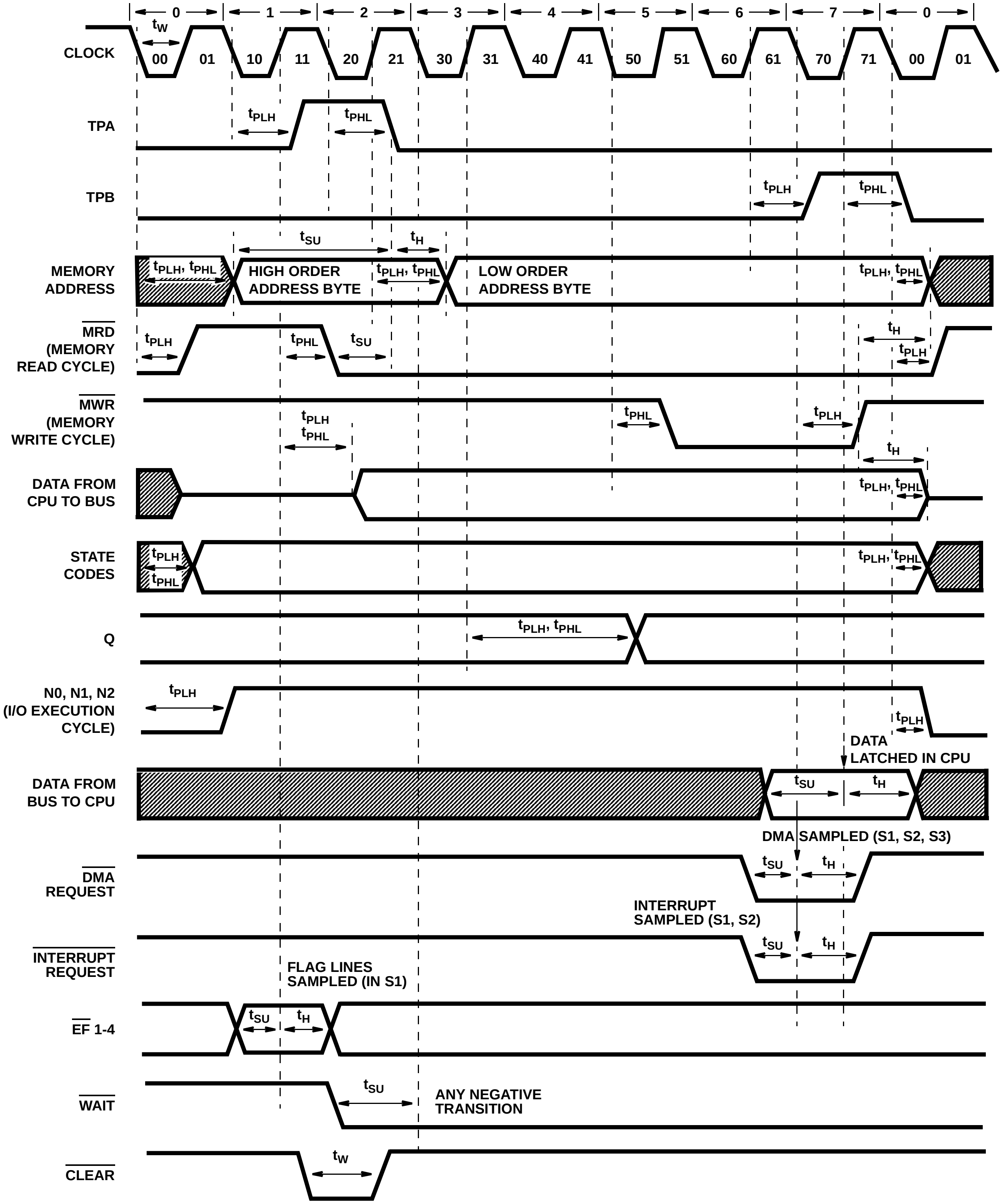


FIGURE 3. BASIC DC TIMING WAVEFORM, ONE INSTRUCTION CYCLE

Timing Waveforms (Continued)



NOTES:

1. This timing diagram is used to show signal relationships only and does not represent any specific machine cycle.
2. All measurements are referenced to 50% point of the waveforms.
3. Shaded areas indicate “Don’t Care” or undefined state. Multiple transitions may occur during this period.

FIGURE 4. TIMING WAVEFORM

Machine Cycle Timing Waveforms (Propagation Delays Not Shown)

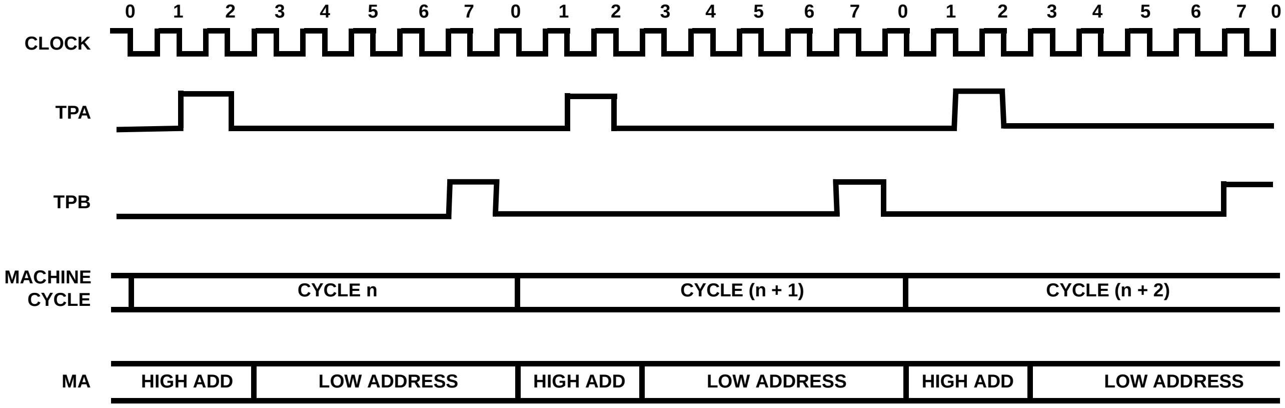


FIGURE 5. GENERAL TIMING WAVEFORMS

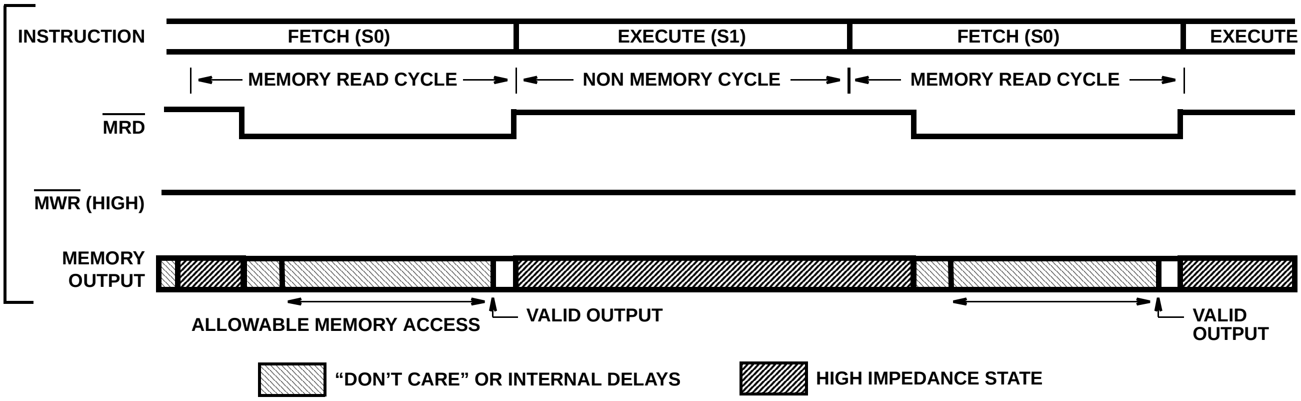


FIGURE 6. NON-MEMORY CYCLE TIMING WAVEFORMS

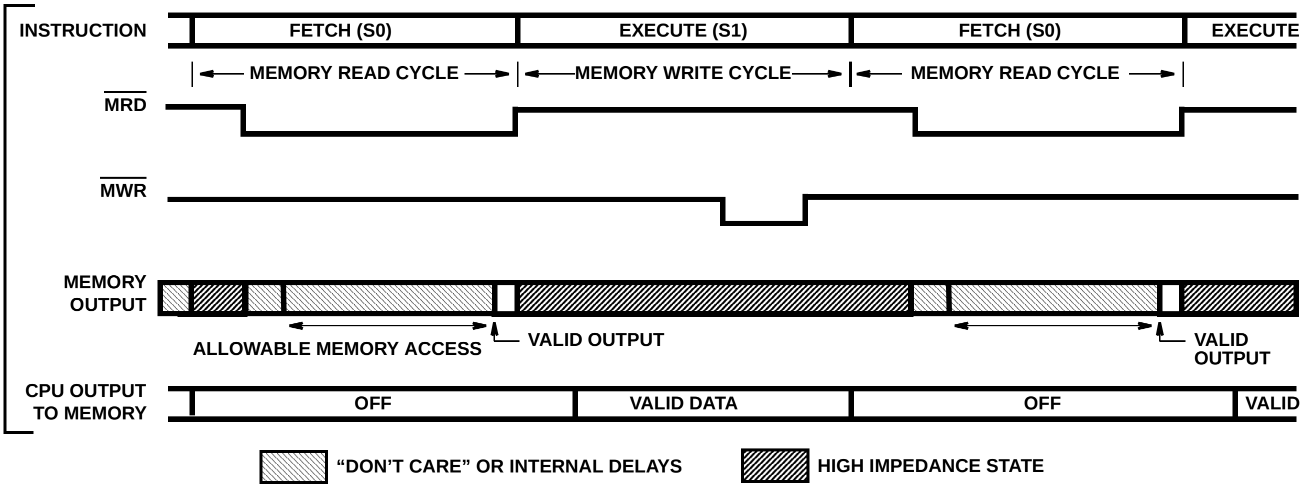


FIGURE 7. MEMORY WRITE CYCLE TIMING WAVEFORMS

Machine Cycle Timing Waveforms (Propagation Delays Not Shown) **(Continued)**

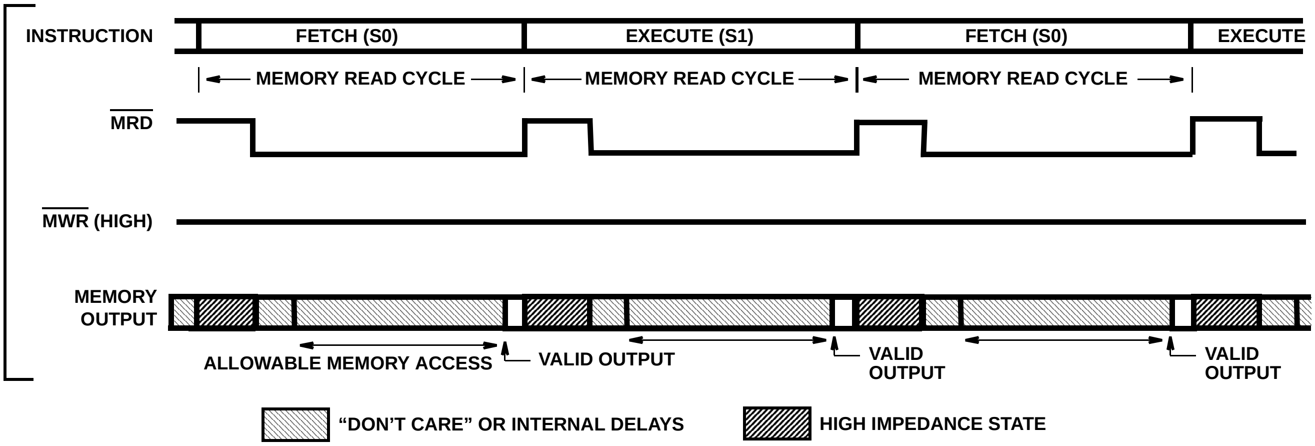


FIGURE 8. MEMORY READ CYCLE TIMING WAVEFORMS

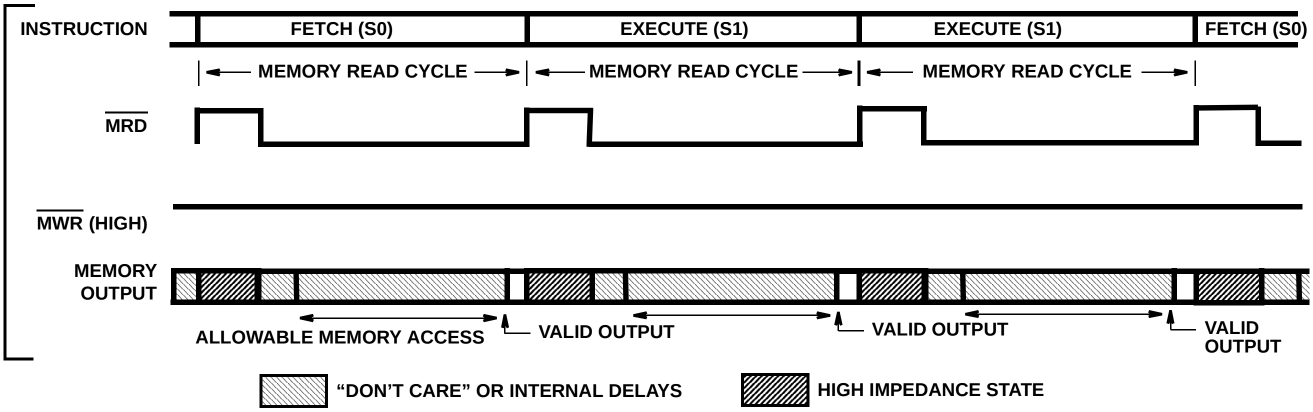


FIGURE 9. LONG BRANCH OR LONG SKIP CYCLE TIMING WAVEFORMS

Machine Cycle Timing Waveforms (Propagation Delays Not Shown) **(Continued)**

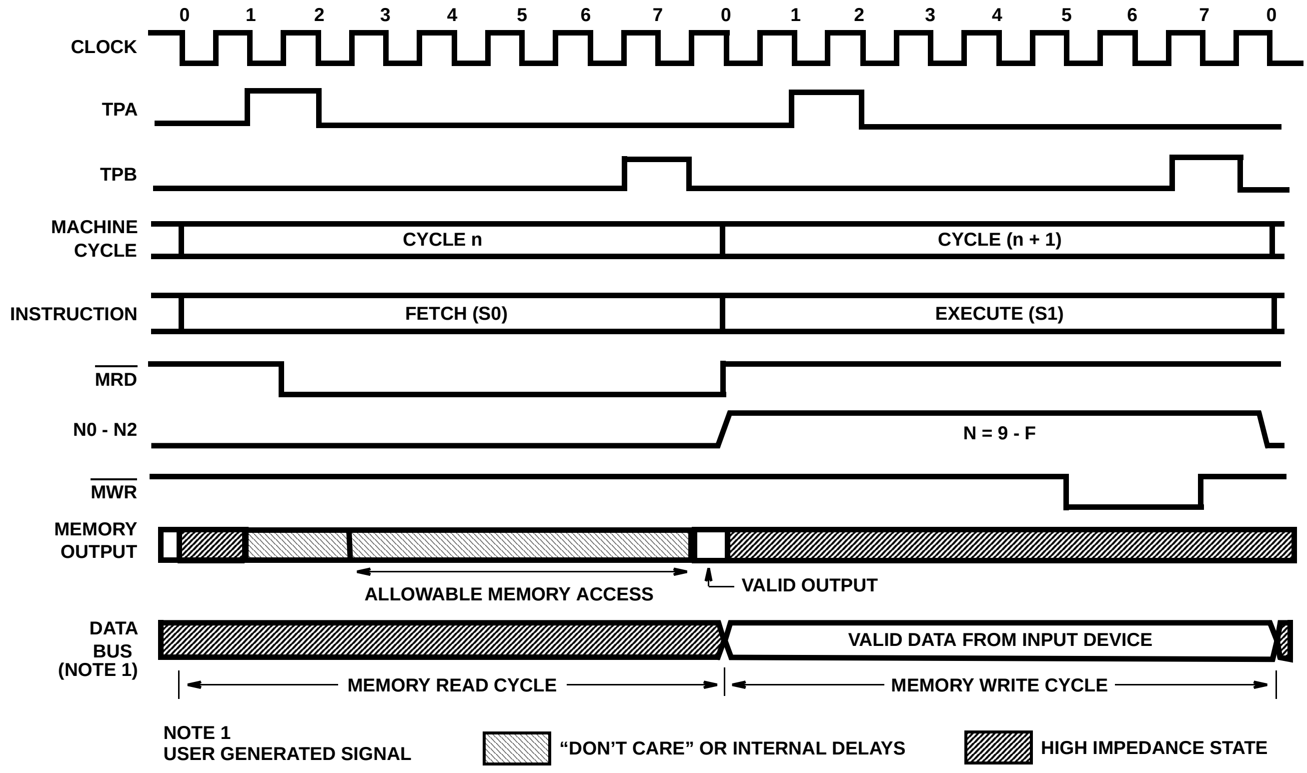


FIGURE 10. INPUT CYCLE TIMING WAVEFORMS

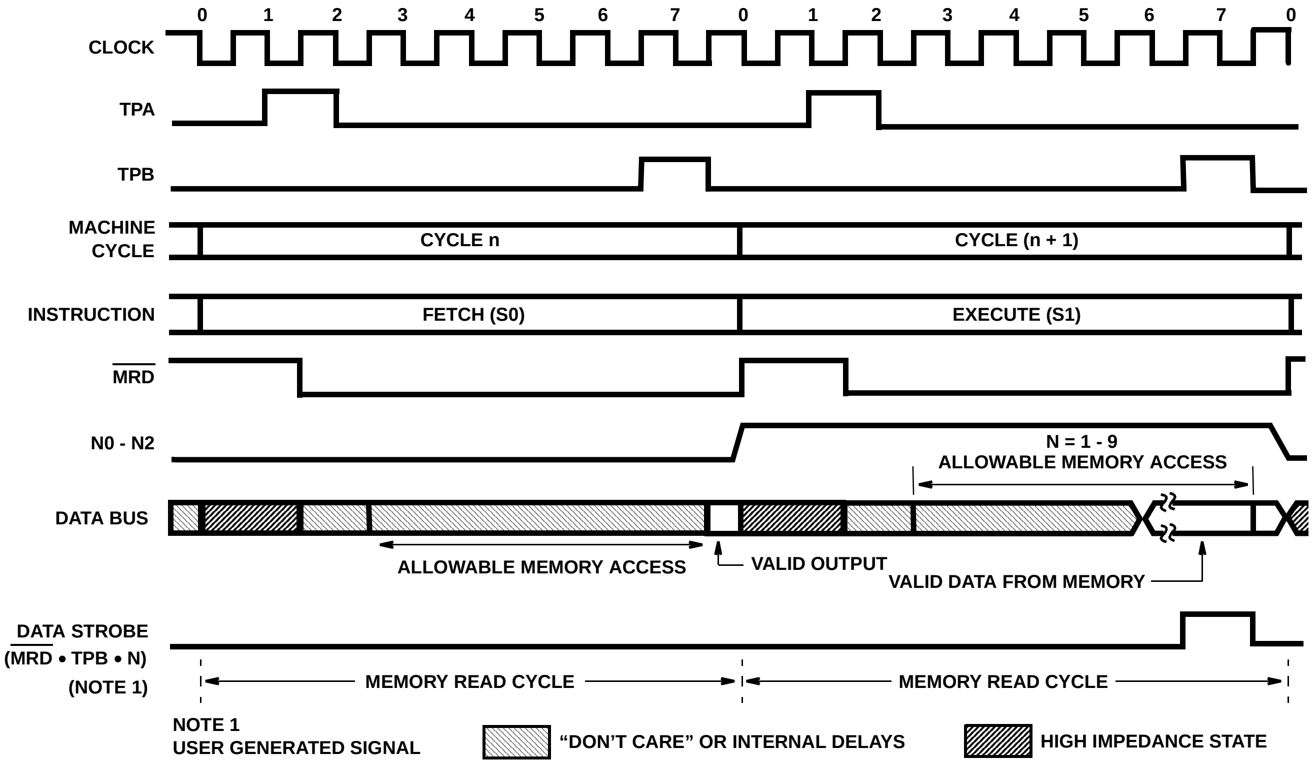


FIGURE 11. OUTPUT CYCLE TIMING WAVEFORMS

Machine Cycle Timing Waveforms (Propagation Delays Not Shown) **(Continued)**

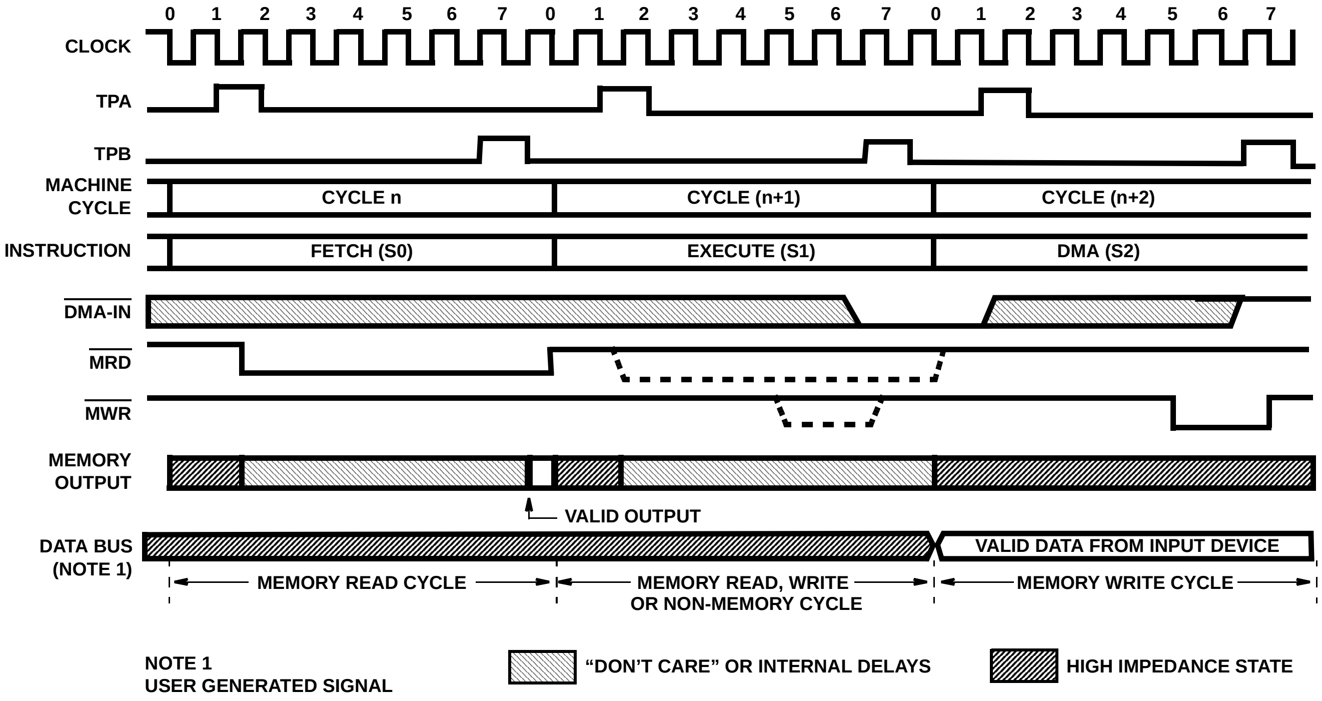


FIGURE 12. $\overline{\text{DMA IN}}$ CYCLE TIMING WAVEFORMS

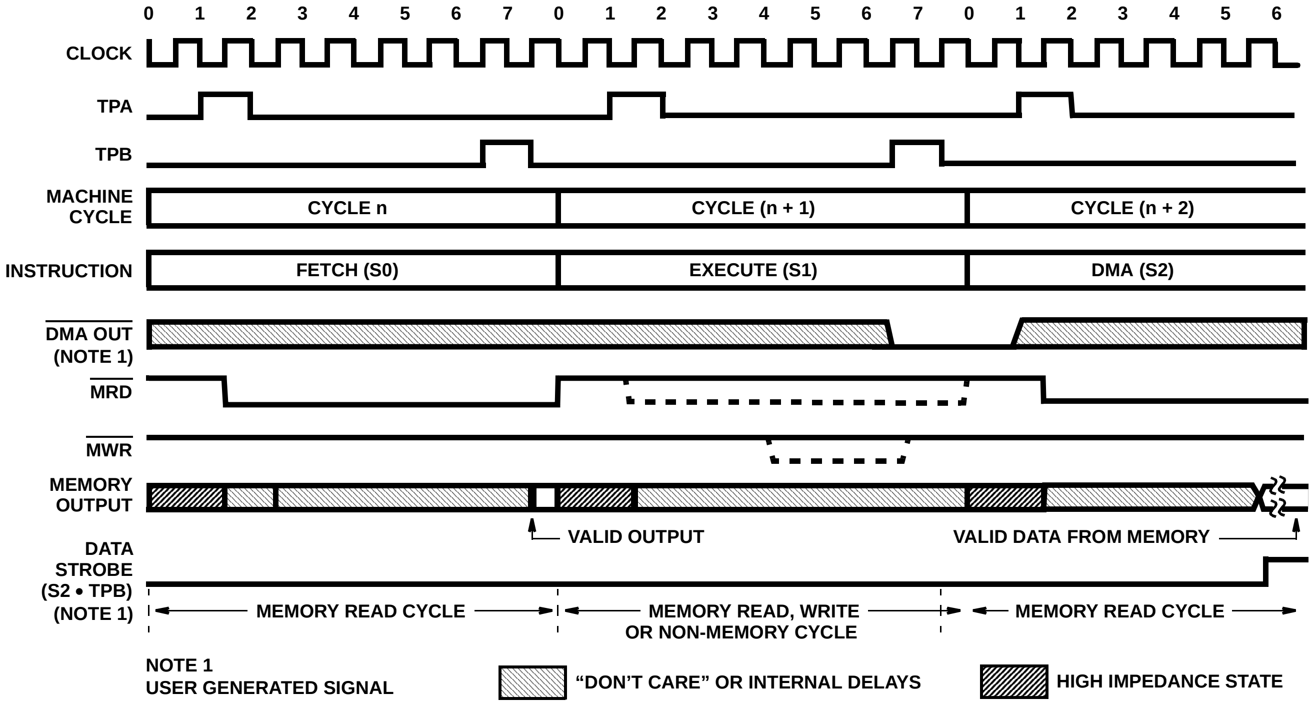


FIGURE 13. $\overline{\text{DMA OUT}}$ CYCLE TIMING WAVEFORMS

Machine Cycle Timing Waveforms (Propagation Delays Not Shown) **(Continued)**

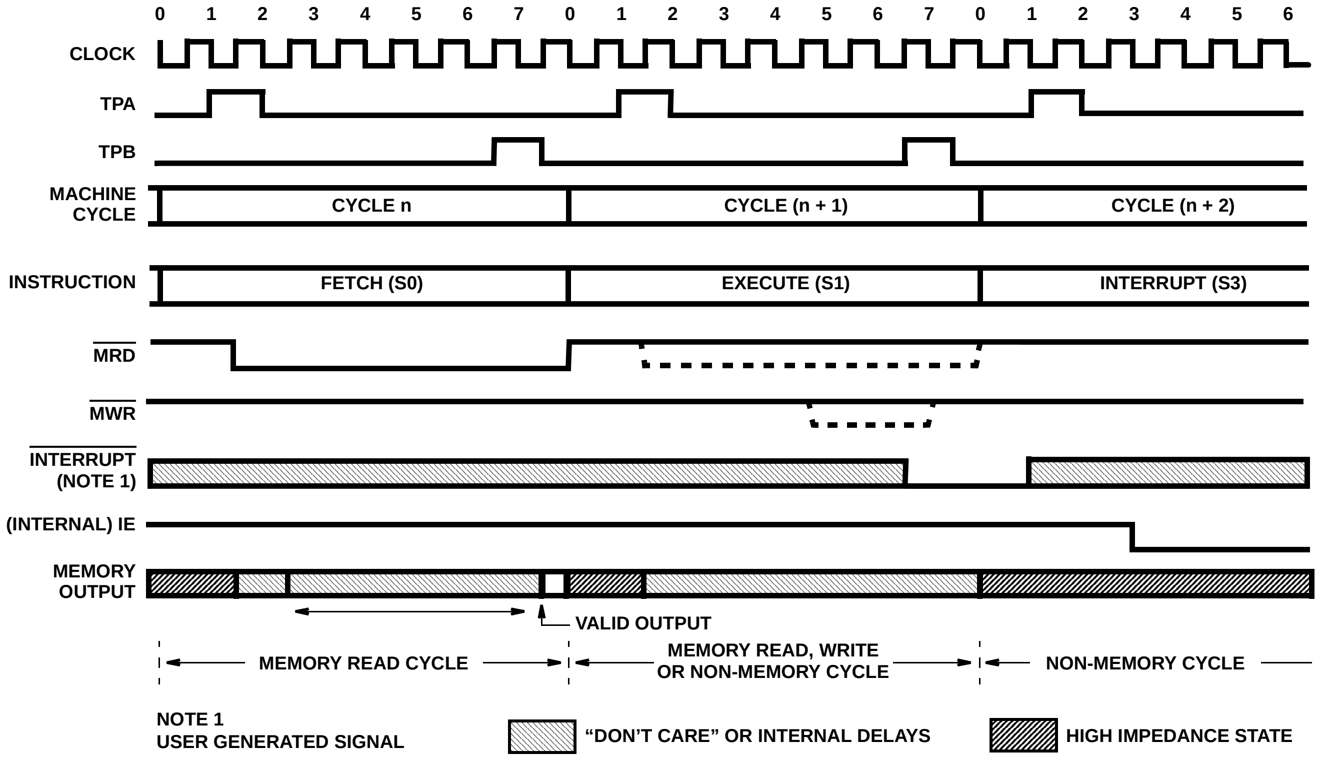


FIGURE 14. INTERRUPT CYCLE TIMING WAVEFORMS

Performance Curves

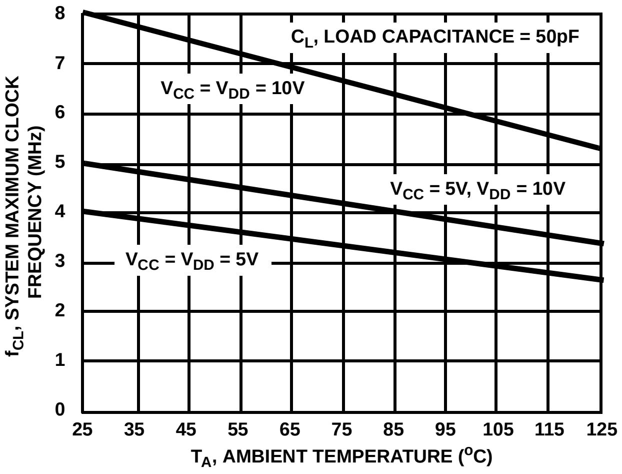


FIGURE 15. CDP1802A, AC TYPICAL MAXIMUM CLOCK FREQUENCY AS A FUNCTION OF TEMPERATURE

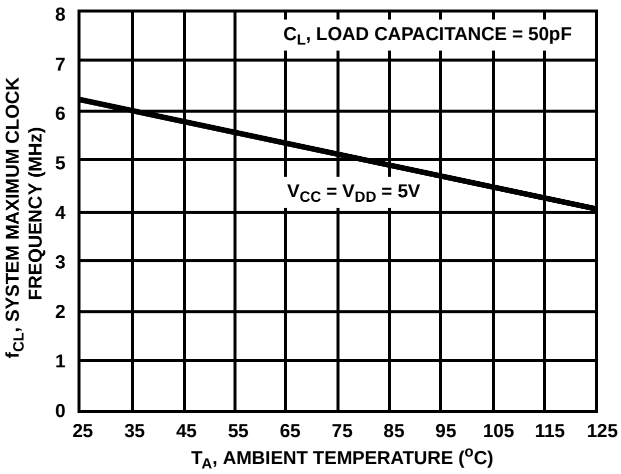


FIGURE 16. CDP1802BC TYPICAL MAXIMUM CLOCK FREQUENCY AS A FUNCTION OF TEMPERATURE

Performance Curves (Continued)

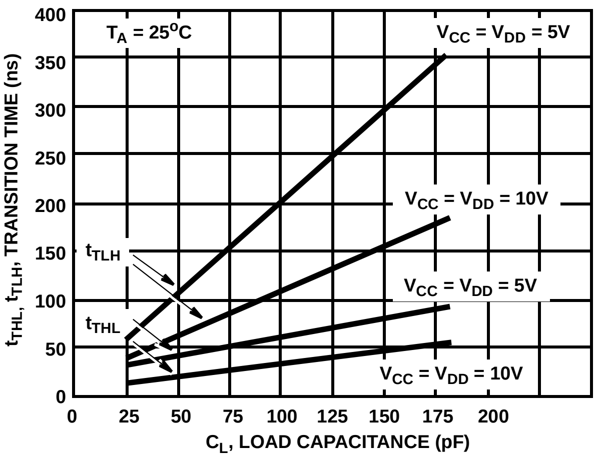


FIGURE 17. TYPICAL TRANSITION TIME vs LOAD CAPACITANCE FOR ALL TYPES

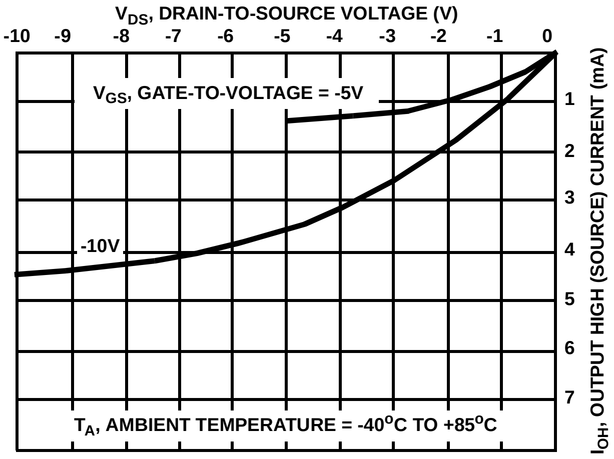


FIGURE 18. CDP1802A, AC MINIMUM OUTPUT HIGH (SOURCE) CURRENT CHARACTERISTICS

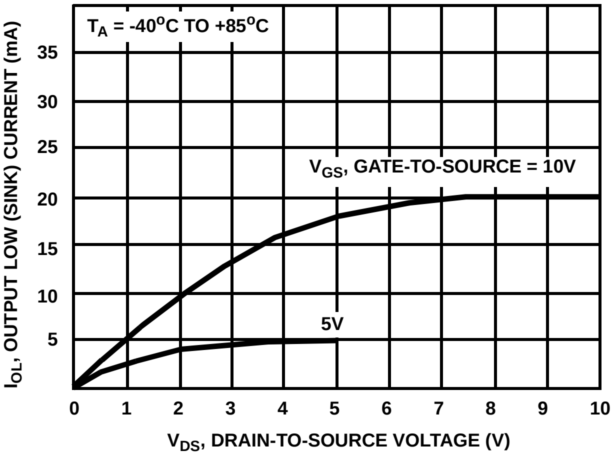


FIGURE 19. CDP1802A, AC MINIMUM OUTPUT LOW (SINK) CURRENT CHARACTERISTICS

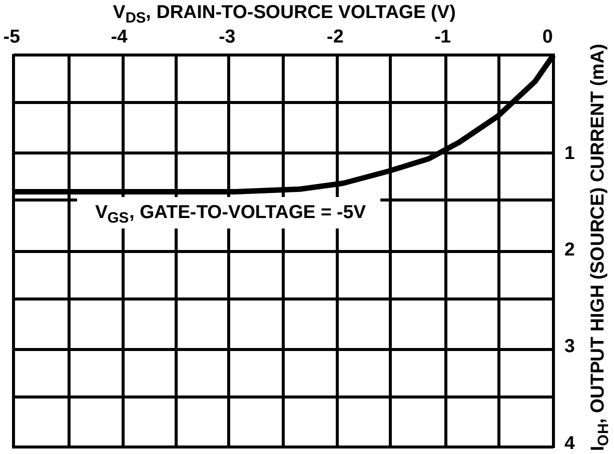


FIGURE 20. CDP1802BC MINIMUM OUTPUT HIGH (SOURCE) CURRENT CHARACTERISTICS

Performance Curves (Continued)

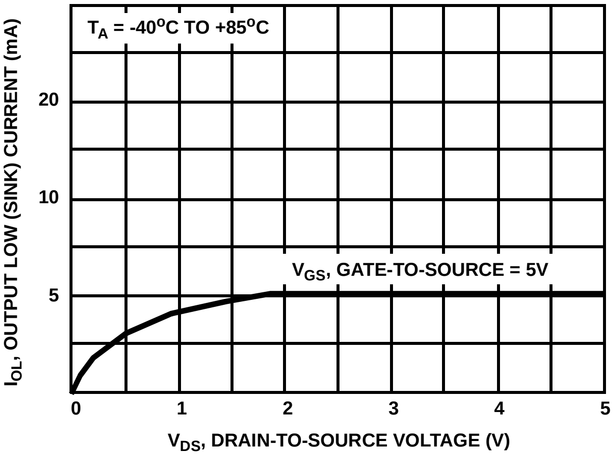
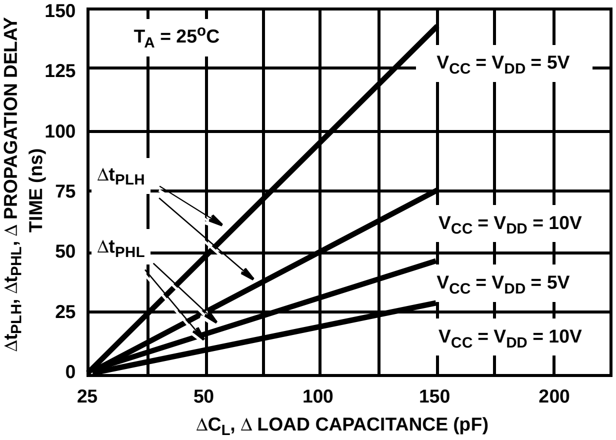
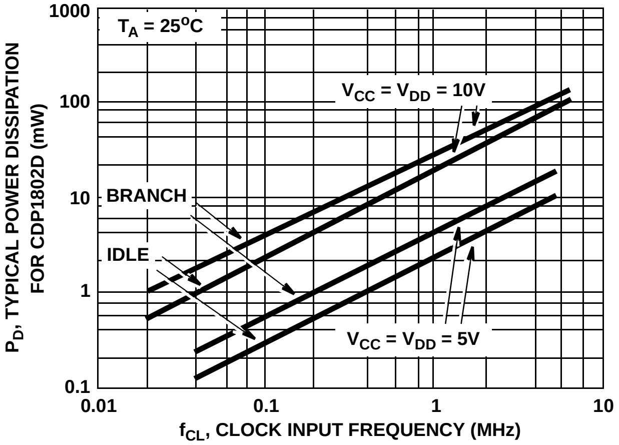


FIGURE 21. CDP1802BC MINIMUM OUTPUT LOW (SINK) CURRENT CHARACTERISTICS



NOTE: ANY OUTPUT EXCEPT $\overline{\text{XTAL}}$

FIGURE 22. TYPICAL CHANGE IN PROPAGATION DELAY AS A FUNCTION OF A CHANGE IN LOAD CAPACITANCE FOR ALL TYPES



NOTE: IDLE = “00” AT M(0000), BRANCH = “3707” AT M(8107), CL = 50pF

FIGURE 23. TYPICAL POWER DISSIPATION AS A FUNCTION OF CLOCK FREQUENCY FOR BRANCH INSTRUCTION AND IDLE INSTRUCTION FOR ALL TYPES

Signal Descriptions

Bus 0 to Bus 7 (Data Bus)

8-bit bidirectional DATA BUS lines. These lines are used for transferring data between the memory, the microprocessor, and I/O devices.

N0 to N2 (I/O Control Lines)

Activated by an I/O instruction to signal the I/O control logic of a data transfer between memory and I/O interface. These lines can be used to issue command codes or device selection codes to the I/O devices (independently or combined with the memory byte on the data bus when an I/O instruction is being executed). The N bits are low at all times except when an I/O instruction is being executed. During this time their state is the same as the corresponding bits in the N register.

The direction of data flow is defined in the I/O instruction by bit N3 (internally) and is indicated by the level of the MRD signal.

$\overline{\text{MRD}} = V_{CC}$: Data from I/O to CPU and Memory

$\overline{\text{MRD}} = V_{SS}$: Data from Memory to I/O

$\overline{\text{EF1}}$ to $\overline{\text{EF4}}$ (4 Flags)

These inputs enable the I/O controllers to transfer status information to the processor. The levels can be tested by the conditional branch instructions. They can be used in conjunction with the INTERRUPT request line to establish interrupt priorities. These flags can also be used by I/O devices to “call the attention” of the processor, in which case the program must routinely test the status of these flag(s). The flag(s) are sampled at the beginning of every S1 cycle.

$\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, $\overline{\text{DMA-OUT}}$ (3 I/O Requests)

These inputs are sampled by the CPU during the interval between the leading edge of TPB and the leading edge of TPA.

Interrupt Action - X and P are stored in T after executing current instruction; designator X is set to 2; designator P is set to 1; interrupt enable is reset to 0 (inhibit); and instruction execution is resumed. The interrupt action requires one machine cycle (S3).

DMA Action - Finish executing current instruction; R(0) points to memory area for data transfer; data is loaded into or read out of memory; and increment R(0).

NOTE: In the event of concurrent DMA and Interrupt requests, DMA-IN has priority followed by DMA-OUT and then Interrupt.

SC0, SC1, (2 State Code Lines)

These outputs indicate that the CPU is: 1) fetching an instruction, or 2) executing an instruction, or 3) processing a DMA request, or 4) acknowledging an interrupt request. The levels of state code are tabulated below. All states are valid at TPA. H = V_{CC}, L = V_{SS}.

STATE TYPE	STATE CODE LINES	
	SC1	SC0
S0 (Fetch)	L	L
S1 (Execute)	L	H
S2 (DMA)	H	L
S3 (Interrupt)	H	H

TPA, TPB (2 Timing Pulses)

Positive pulses that occur once in each machine cycle (TPB follows TPA). They are used by I/O controllers to interpret codes and to time interaction with the data bus. The trailing edge of TPA is used by the memory system to latch the higher-order byte of the 16-bit memory address. TPA is suppressed in IDLE when the CPU is in the load mode.

MA0 to MA7 (8 Memory Address Lines)

In each cycle, the higher-order byte of a 16-bit CPU memory address appears on the memory address lines MA0-7 first. Those bits required by the memory system can be strobed into external address latches by timing pulse TPA. The low order byte of the 16-bit address appears on the address lines after the termination of TPA. Latching of all 8 higher-order address bits would permit a memory system of 64K bytes.

$\overline{\text{MWR}}$ (Write Pulse)

A negative pulse appearing in a memory-write cycle, after the address lines have stabilized.

$\overline{\text{MRD}}$ (Read Level)

A low level on MRD indicates a memory read cycle. It can be used to control three-state outputs from the addressed memory which may have a common data input and output bus. If a memory does not have a three-state high-impedance output, MRD is useful for driving memory/bus separator gates. It is also used to indicate the direction of data transfer during an I/O instruction. For additional information see Table 1.

Q

Single bit output from the CPU which can be set or reset under program control. During SEQ or REQ instruction execution, Q is set or reset between the trailing edge of TPA and the leading edge of TPB.

CLOCK

Input for externally generated single-phase clock. The clock is counted down internally to 8 clock pulses per machine cycle.

$\overline{\text{XTAL}}$

Connection to be used with clock input terminal, for an external crystal, if the on-chip oscillator is utilized. The crystal is connected between terminals 1 and 39 (CLOCK and $\overline{\text{XTAL}}$) in parallel with a resistance (10MΩ typ). Frequency trimming capacitors may be required at terminals 1 and 39. For additional information, see Application Note AN6565.

$\overline{\text{WAIT}}$, $\overline{\text{CLEAR}}$ (2 Control Lines)

Provide four control modes as listed in the following truth table:

$\overline{\text{CLEAR}}$	$\overline{\text{WAIT}}$	MODE
L	L	LOAD
L	H	RESET
H	L	PAUSE
H	H	RUN

V_{DD}, V_{SS}, V_{CC} (Power Levels)

The internal voltage supply V_{DD} is isolated from the Input/Output voltage supply V_{CC} so that the processor may operate at maximum speed while interfacing with peripheral devices operating at lower voltage. V_{CC} must be less than or equal to V_{DD}. All outputs swing from V_{SS} to V_{CC}. The recommended input voltage swing is V_{SS} to V_{CC}.

Architecture

The CPU block diagram is shown in Figure 2. The principal feature of this system is a register array (R) consisting of sixteen 16-bit scratchpad registers. Individual registers in the array (R) are designated (selected) by a 4-bit binary code from one of the 4-bit registers labeled N, P and X. The contents of any register can be directed to any one of the following three paths:

1. The external memory (multiplexed, higher-order byte first, on to 8 memory address lines).
2. The D register (either of the two bytes can be gated to D).
3. The increment/decrement circuit where it is increased or decreased by one and stored back in the selected 16-bit register.

The three paths, depending on the nature of the instruction, may operate independently or in various combinations in the same machine cycle.

With two exceptions, CPU instruction consists of two 8-clock-pulse machine cycles. The first cycle is the fetch cycle, and the second - and third if necessary - are execute cycles. During the fetch cycle the four bits in the P designator select one of the 16 registers R(P) as the current program counter. The selected register R(P) contains the address of the memory location from which the instruction is to be fetched. When the instruction is read out from the memory, the higher order 4 bits of the instruction byte are loaded into the register and the lower order 4 bits into the N register. The content of the program counter is automatically incremented by one so that R(P) is now "pointing" to the next byte in the memory.

The X designator selects one of the 16 registers R(X) to "point" to the memory for an operand (or data) in certain ALU or I/O operations.

The N designator can perform the following five functions depending on the type of instruction fetched:

1. Designate one of the 16 registers in R to be acted upon during register operations.
2. Indicate to the I/O devices a command code or device selection code for peripherals.
3. Indicate the specific operation to be executed during the ALU instructions, types of test to be performed during the Branch instruction, or the specific operation required in a class of miscellaneous instructions (70 - 73 and 78 - 7B).
4. Indicate the value to be loaded into P to designate a new register to be used as the program counter R(P).
5. Indicate the value to be loaded into X to designate a new register to be used as data pointer R(X).

The registers in R can be assigned by a programmer in three different ways: as program counters, as data pointers, or as scratchpad locations (data registers) to hold two bytes of data.

Program Counters

Any register can be the main program counter; the address of the selected register is held in the P designator. Other registers in R can be used as subroutine program counters. By single instruction the contents of the P register can be changed to effect a "call" to a subroutine. When interrupts are being serviced, register R(1) is used as the program counter for the user's interrupt servicing routine. After reset, and during a DMA operation, R(0) is used as the program counter. At all other times the register designated as program counter is at the discretion of the user.

Data Pointers

The registers in R may be used as data pointers to indicate a location in memory. The register designated by X (i.e., R(X)) points to memory for the following instructions (see Table 1).

1. ALU operations F1 - F5, F7, 74, 75, 77
2. Output instructions 61 through 67
3. Input instructions 69 through 6F
4. Certain miscellaneous instructions - 70 - 73, 78, 60, F0

The register designated by N (i.e., R(N)) points to memory for the "load D from memory" instructions 0N and 4N and the "Store D" instruction 5N. The register designated by P (i.e., the program counter) is used as the data pointer for ALU instructions F8 - FD, FF, 7C, 7D, 7F. During these instruction executions, the operation is referred to as "data immediate".

Another important use of R as a data pointer supports the built-in Direct-Memory-Access (DMA) function. When a DMA-In or DMA-Out request is received, one machine cycle is "stolen". This operation occurs at the end of the execute machine cycle in the current instruction. Register R(0) is always used as the data pointer during the DMA operation. The data is read from (DMA-Out) or written into (DMA-In) the memory location pointed to by the R(0) register. At the end of the transfer, R(0) is incremented by one so that the processor is ready to act upon the next DMA byte transfer request. This feature in the 1800-series architecture saves a substantial amount of logic when fast exchanges of blocks of data are required, such as with magnetic discs or during CRT-display-refresh cycles.

Data Registers

When registers in R are used to store bytes of data, four instructions are provided which allow D to receive from or write into either the higher-order or lower-order byte portions of the register designated by N. By this mechanism (together with loading by data immediate) program pointer and data pointer designations are initialized. Also, this technique allows scratchpad registers in R to be used to hold general data. By employing increment or decrement instructions, such registers may be used as loop counters.

The Q Flip-Flop

An internal flip-flop, Q, can be set or reset by instruction and can be sensed by conditional branch instructions. The output of Q is also available as a microprocessor output.

Interrupt Servicing

Register R(1) is always used as the program counter whenever interrupt servicing is initiated. When an interrupt request occurs and the interrupt is allowed by the program (again, nothing takes place until the completion of the current instruction), the contents of the X and P registers are stored in the temporary register T, and X and P are set to new values; hex digit 2 in X and hex digit 1 in P. Interrupt Enable is automatically deactivated to inhibit further interrupts. The user's interrupt routine is now in control; the contents of T may be saved by means of a single instruction (78) in the memory location pointed to by R(X). At the conclusion of the interrupt, the user's routine may restore the pre-interrupted value of X and P with a single instruction (70 or 71). The Interrupt Enable flip-flop can be activated to permit further interrupts or can be disabled to prevent them.

CPU Register Summary

D	8 Bits	Data Register (Accumulator)
DF	1-Bit	Data Flag (ALU Carry)
B	8 Bits	Auxiliary Holding Register
R	16 Bits	1 of 16 Scratchpad Registers
P	4 Bits	Designates which register is Program Counter
X	4 Bits	Designates which register is Data Pointer
N	4 Bits	Holds Low-Order Instruction Digit
I	4 Bits	Holds High-Order Instruction Digit
T	8 Bits	Holds old X, P after Interrupt (X is high nibble)
IE	1-Bit	Interrupt Enable
Q	1-Bit	Output Flip-Flop

CDP1802 Control Modes

The $\overline{\text{WAIT}}$ and $\overline{\text{CLEAR}}$ lines provide four control modes as listed in the following truth table:

CLEAR	WAIT	MODE
L	L	LOAD
L	H	RESET
H	L	PAUSE
H	H	RUN

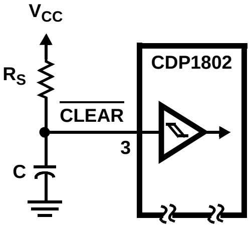
The function of the modes are defined as follows:

Load

Holds the CPU in the IDLE execution state and allows an I/O device to load the memory without the need for a “bootstrap” loader. It modifies the IDLE condition so that DMA-IN operation does not force execution of the next instruction.

Reset

Registers I, N, Q are reset, IE is set and 0’s (VSS) are placed on the data bus. TPA and TPB are suppressed while reset is held and the CPU is placed in S1. The first machine cycle after termination of reset is an initialization cycle which requires 9 clock pulses. During this cycle the CPU remains in S1 and register X, P, and R(0) are reset. Interrupt and DMA servicing are suppressed during the initialization cycle. The next cycle is an S0, S1, or an S2 but never an S3. With the use of a 71 instruction followed by 00 at memory locations 0000 and 0001, this feature may be used to reset IE, so as to preclude interrupts until ready for them. Power-up reset can be realized by connecting an RC network directly to the CLEAR pin, since it has a Schmitt triggered input, see Figure 24.



THE RC TIME CONSTANT SHOULD BE GREATER THAN THE OSCILLATOR START-UP TIME (TYPICALLY 20ms)

FIGURE 24. RESET DIAGRAM

Pause

Stops the internal CPU timing generator on the first negative high-to-low transition of the input clock. The oscillator continues to operate, but subsequent clock transitions are ignored.

Run

May be initiated from the Pause or Reset mode functions. If initiated from Pause, the CPU resumes operation on the first negative high-to-low transition of the input clock. When initiated from the Reset operation, the first machine cycle following Reset is always the initialization cycle. The initialization cycle is then followed by a DMA (S2) cycle or fetch (S0) from location 0000 in memory.

Run-Mode State Transitions

The CPU state transitions when in the RUN and RESET modes are shown in Figure 25. Each machine cycle requires the same period of time, 8 clock pulses, except the initialization cycle, which requires 9 clock pulses. The execution of an instruction requires either two or three machine cycles, S0 followed by a single S1 cycle or two S1 cycles. S2 is the response to a DMA request and S3 is the interrupt response. Table 2 shows the conditions on Data Bus and Memory Address lines during all machine states.

Instruction Set

The CPU instruction summary is given in Table 1. Hexadecimal notation is used to refer to the 4-bit binary codes.

In all registers bits are numbered from the least significant bit (LSB) to the most significant bit (MSB) starting with 0.

R(W): Register designated by W, where

W = N or X, or P

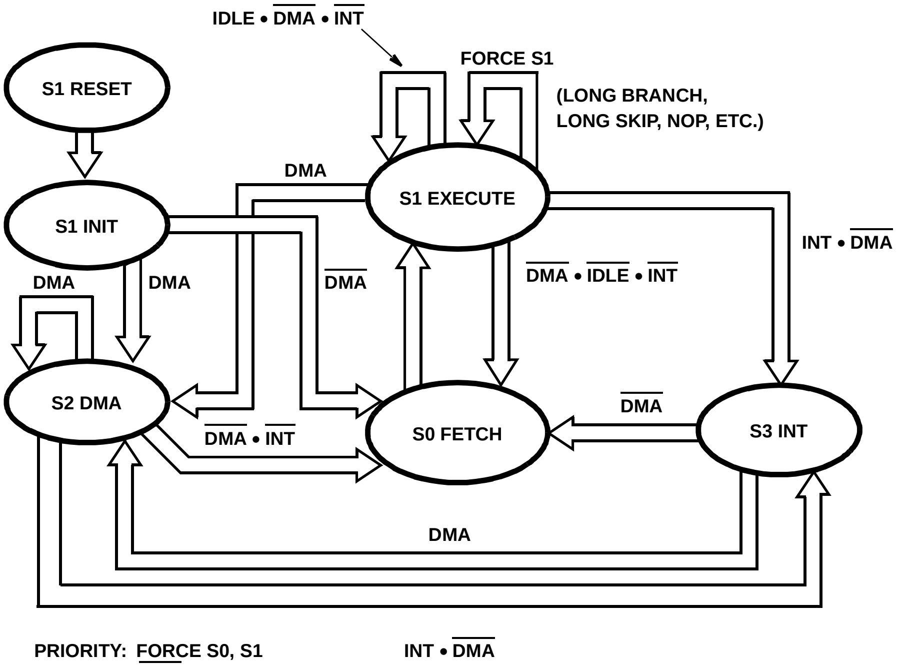
R(W).0: Lower order byte of R(W)

R(W).1: Higher order byte of R(W)

Operation Notation

M(R(N)) → D; R(N) + 1 → R(N)

This notation means: The memory byte pointed to by R(N) is loaded into D, and R(N) is incremented by 1.


IDLE • $\overline{DMA}$ • $\overline{INT}$
FORCE S1
(LONG BRANCH, LONG SKIP, NOP, ETC.)
S1 RESET
S1 INIT
S1 EXECUTE
DMA
DMA
DMA
$\overline{DMA}$
$\overline{DMA}$ • $\overline{IDLE}$ • $\overline{INT}$
INT • $\overline{DMA}$
S2 DMA
S0 FETCH
S3 INT
$\overline{DMA}$
$\overline{DMA}$ • $\overline{INT}$
DMA
INT • $\overline{DMA}$

PRIORITY: FORCE S0, S1
$\overline{DMA}$ IN
$\overline{DMA}$ OUT
$\overline{INT}$

FIGURE 25. STATE TRANSITION DIAGRAM

TABLE 1. INSTRUCTION SUMMARY (SEE NOTES)

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
MEMORY REFERENCE			
LOAD VIA N	LDN	0N	M(R(N)) → D; FOR N not 0
LOAD ADVANCE	LDA	4N	M(R(N)) → D; R(N) + 1 → R(N)
LOAD VIA X	LDX	F0	M(R(X)) → D
LOAD VIA X AND ADVANCE	LDXA	72	M(R(X)) → D; R(X) + 1 → R(X)
LOAD IMMEDIATE	LDI	F8	M(R(P)) → D; R(P) + 1 → R(P)
STORE VIA N	STR	5N	D → M(R(N))
STORE VIA X AND DECREMENT	STXD	73	D → M(R(X)); R(X) - 1 → R(X)
REGISTER OPERATIONS			
INCREMENT REG N	INC	1N	R(N) + 1 → R(N)
DECREMENT REG N	DEC	2N	R(N) - 1 → R(N)
INCREMENT REG X	IRX	60	R(X) + 1 → R(X)
GET LOW REG N	GLO	8N	R(N).0 → D
PUT LOW REG N	PLO	AN	D → R(N).0
GET HIGH REG N	GHI	9N	R(N).1 → D
PUT HIGH REG N	PHI	BN	D → R(N).1
LOGIC OPERATIONS (Note 1)			
OR	OR	F1	M(R(X)) OR D → D
OR IMMEDIATE	ORI	F9	M(R(P)) OR D → D; R(P) + 1 → R(P)

TABLE 1. INSTRUCTION SUMMARY (SEE NOTES) **(Continued)**

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
EXCLUSIVE OR	XOR	F3	M(R(X)) XOR D → D
EXCLUSIVE OR IMMEDIATE	XRI	FB	M(R(P)) XOR D → D; R(P) + 1 → R(P)
AND	AND	F2	M(R(X)) AND D → D
AND IMMEDIATE	ANI	FA	M(R(P)) AND D → D; R(P) + 1 → R(P)
SHIFT RIGHT	SHR	F6	SHIFT D RIGHT, LSB(D) → DF, 0 → MSB(D)
SHIFT RIGHT WITH CARRY	SHRC	76 (Note 2)	SHIFT D RIGHT, LSB(D) → DF, DF → MSB(D)
RING SHIFT RIGHT	RSHR	76 (Note 2)	SHIFT D RIGHT, LSB(D) → DF, DF → MSB(D)
SHIFT LEFT	SHL	FE	SHIFT D LEFT, MSB(D) → DF, 0 → LSB(D)
SHIFT LEFT WITH CARRY	SHLC	7E (Note 2)	SHIFT D LEFT, MSB(D) → DF, DF → LSB(D)
RING SHIFT LEFT	RSHL	7E (Note 2)	SHIFT D LEFT, MSB(D) → DF, DF → LSB(D)
ARITHMETIC OPERATIONS (Note 1)			
ADD	ADD	F4	M(R(X)) + D → DF, D
ADD IMMEDIATE	ADI	FC	M(R(P)) + D → DF, D; R(P) + 1 → R(P)
ADD WITH CARRY	ADC	74	M(R(X)) + D + DF → DF, D
ADD WITH CARRY, IMMEDIATE	ADCI	7C	M(R(P)) + D + DF → DF, D; R(P) + 1 → R(P)
SUBTRACT D	SD	F5	M(R(X)) - D → DF, D
SUBTRACT D IMMEDIATE	SDI	FD	M(R(P)) - D → DF, D; R(P) + 1 → R(P)
SUBTRACT D WITH BORROW	SDB	75	M(R(X)) - D - (NOT DF) → DF, D
SUBTRACT D WITH BORROW, IMMEDIATE	SDBI	7D	M(R(P)) - D - (Not DF) → DF, D; R(P) + 1 → R(P)
SUBTRACT MEMORY	SM	F7	D-M(R(X)) → DF, D
SUBTRACT MEMORY IMMEDIATE	SMI	FF	D-M(R(P)) → DF, D; R(P) + 1 → R(P)
SUBTRACT MEMORY WITH BORROW	SMB	77	D-M(R(X))-(NOT DF) → DF, D
SUBTRACT MEMORY WITH BORROW, IMMEDIATE	SMBI	7F	D-M(R(P))-(NOT DF) → DF, D; R(P) + 1 → R(P)
BRANCH INSTRUCTIONS - SHORT BRANCH			
SHORT BRANCH	BR	30	M(R(P)) → R(P).0
NO SHORT BRANCH (See SKP)	NBR	38 (Note 2)	R(P) + 1 → R(P)
SHORT BRANCH IF D = 0	BZ	32	IF D = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF D NOT 0	BNZ	3A	IF D NOT 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF DF = 1	BDF	33 (Note 2)	IF DF = 1, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF POS OR ZERO	BPZ		
SHORT BRANCH IF EQUAL OR GREATER	BGE		
SHORT BRANCH IF DF = 0	BNF	3B (Note 2)	IF DF = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF MINUS	BM		
SHORT BRANCH IF LESS	BL		
SHORT BRANCH IF Q = 1	BQ	31	IF Q = 1, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF Q = 0	BNQ	39	IF Q = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)

TABLE 1. INSTRUCTION SUMMARY (SEE NOTES) **(Continued)**

INSTRUCTION	**MNEMONIC**	**OP CODE**	**OPERATION**
SHORT BRANCH IF EF1 = 1 ($\overline{\text{EF1}}$ = V_{SS})	B1	34	IF EF1 =1, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF1 = 0 ($\overline{\text{EF1}}$ = V_{CC})	BN1	3C	IF EF1 = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF2 = 1 ($\overline{\text{EF2}}$ = V_{SS})	B2	35	IF EF2 = 1, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF2 = 0 ($\overline{\text{EF2}}$ = V_{CC})	BN2	3D	IF EF2 = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF3 = 1 ($\overline{\text{EF3}}$ = V_{SS})	B3	36	IF EF3 = 1, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF3 = 0 ($\overline{\text{EF3}}$ = V_{CC})	BN3	3E	IF EF3 = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF4 = 1 ($\overline{\text{EF4}}$ = V_{SS})	B4	37	IF EF4 = 1, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
SHORT BRANCH IF EF4 = 0 ($\overline{\text{EF4}}$ = V_{CC})	BN4	3F	IF EF4 = 0, M(R(P)) → R(P).0, ELSE R(P) + 1 → R(P)
BRANCH INSTRUCTIONS - LONG BRANCH			
LONG BRANCH	LBR	C0	M(R(P)) → R(P). 1, M(R(P) + 1) → R(P).0
NO LONG BRANCH (See LSKP)	NLBR	C8 (Note 2)	R(P) + 2 → R(P)
LONG BRANCH IF D = 0	LBZ	C2	IF D = 0, M(R(P)) → R(P).1, M(R(P) +1) → R(P).0, ELSE R(P) + 2 → R(P)
LONG BRANCH IF D NOT 0	LBNZ	CA	IF D Not 0, M(R(P)) → R(P).1, M(R(P) + 1) → R(P).0, ELSE R(P) + 2 → R(P)
LONG BRANCH IF DF = 1	LBDF	C3	IF DF = 1, M(R(P)) → R(P).1, M(R(P) + 1) → R(P).0, ELSE R(P) + 2 → R(P)
LONG BRANCH IF DF = 0	LBNF	CB	IF DF = 0, M(R(P)) → R(P).1, M(R(P) + 1) → R(P).0, ELSE R(P) + 2 → R(P)
LONG BRANCH IF Q = 1	LBQ	C1	IF Q = 1, M(R(P)) → R(P).1, M(R(P) + 1) → R(P).0, ELSE R(P) + 2 → R(P)
LONG BRANCH IF Q = 0	LBNQ	C9	IF Q = 0, M(R(P)) → R(P).1, M(R(P) + 1) → R(P).0 EISE R(P) + 2 → R(P)
SKIP INSTRUCTIONS			
SHORT SKIP (See NBR)	SKP	38 (Note 2)	R(P) + 1 → R(P)
LONG SKIP (See NLBR)	LSKP	C8 (Note 2)	R(P) + 2 → R(P)
LONG SKIP IF D = 0	LSZ	CE	IF D = 0, R(P) + 2 → R(P), ELSE CONTINUE
LONG SKIP IF D NOT 0	LSNZ	C6	IF D Not 0, R(P) + 2 → R(P), ELSE CONTINUE
LONG SKIP IF DF = 1	LSDF	CF	IF DF = 1, R(P) + 2 → R(P), ELSE CONTINUE
LONG SKIP IF DF = 0	LSNF	C7	IF DF = 0, R(P) + 2 → R(P), ELSE CONTINUE
LONG SKIP IF Q = 1	LSQ	CD	IF Q = 1, R(P) + 2 → R(P), ELSE CONTINUE
LONG SKIP IF Q = 0	LSNQ	C5	IF Q = 0, R(P) + 2 → R(P), ELSE CONTINUE
LONG SKIP IF IE = 1	LSIE	CC	IF IE = 1, R(P) + 2 → R(P), ELSE CONTINUE
CONTROL INSTRUCTIONS			
IDLE	IDL	00 (Note 3)	WAIT FOR DMA OR INTERRUPT; M(R(0)) → BUS
NO OPERATION	NOP	C4	CONTINUE
SET P	SEP	DN	N → P
SET X	SEX	EN	N → X
SET Q	SEQ	7B	1 → Q

TABLE 1. INSTRUCTION SUMMARY (SEE NOTES) **(Continued)**

INSTRUCTION	MNEMONIC	OP CODE	OPERATION
RESET Q	REQ	7A	0 → Q
SAVE	SAV	78	T → M(R(X))
PUSH X, P TO STACK	MARK	79	(X, P) → T; (X, P) → M(R(2)), THEN P → X; R(2) - 1 → R(2)
RETURN	RET	70	M(R(X)) → (X, P); R(X) + 1 → R(X), 1 → IE
DISABLE	DIS	71	M(R(X)) → (X, P); R(X) + 1 → R(X), 0 → IE
INPUT - OUTPUT BYTE TRANSFER			
OUTPUT 1	OUT 1	61	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 1
OUTPUT 2	OUT 2	62	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 2
OUTPUT 3	OUT 3	63	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 3
OUTPUT 4	OUT 4	64	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 4
OUTPUT 5	OUT 5	65	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 5
OUTPUT 6	OUT 6	66	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 6
OUTPUT 7	OUT 7	67	M(R(X)) → BUS; R(X) + 1 → R(X); N LINES = 7
INPUT 1	INP 1	69	BUS → M(R(X)); BUS → D; N LINES = 1
INPUT 2	INP 2	6A	BUS → M(R(X)); BUS → D; N LINES = 2
INPUT 3	INP 3	6B	BUS → M(R(X)); BUS → D; N LINES = 3
INPUT 4	INP 4	6C	BUS → M(R(X)); BUS → D; N LINES = 4
INPUT 5	INP 5	6D	BUS → M(R(X)); BUS → D; N LINES = 5
INPUT 6	INP 6	6E	BUS → M(R(X)); BUS → D; N LINES = 6
INPUT 7	INP 7	6F	BUS → M(R(X)); BUS → D; N LINES = 7

TABLE 1. INSTRUCTION SUMMARY (SEE NOTES) **(Continued)**

INSTRUCTION	MNEMONIC	OP CODE	OPERATION

NOTES: (For Table 1)

1. The arithmetic operations and the shift instructions are the only instructions that can alter the DF.

   After an add instruction:
   DF = 1 denotes a carry has occurred
   DF = 0 Denotes a carry has not occurred
   After a subtract instruction:
   DF = 1 denotes no borrow. D is a true positive number
   DF = 0 denotes a borrow. D is two's complement
   The syntax "-(not DF)" denotes the subtraction of the borrow.

2. This instruction is associated with more than one mnemonic. Each mnemonic is individually listed.

3. An idle instruction initiates a repeating S1 cycle. The processor will continue to idle until an I/O request ($\overline{\text{INTERRUPT}}$, $\overline{\text{DMA-IN}}$, or $\overline{\text{DMA- OUT}}$) is activated. When the request is acknowledged, the idle cycle is terminated and the I/O request is serviced, and then normal operation is resumed.

4. Long-Branch, Long-Skip and No Op instructions require three cycles to complete (1 fetch + 2 execute).

   Long-Branch instructions are three bytes long. The first byte specifies the condition to be tested; and the second and third byte, the branching address.

   The long-branch instructions can:
   a. Branch unconditionally
   b. Test for D = 0 or D ≠ 0
   c. Test for DF = 0 or DF = 1
   d. Test for Q = 0 or Q = 1
   e. Effect an unconditional no branch

   If the tested condition is met, then branching takes place; the branching address bytes are loaded in the high-and-low order bytes of the current program counter, respectively. This operation effects a branch to any memory location.

   If the tested condition is not met, the branching address bytes are skipped over, and the next instruction in sequence is fetched and executed. This operation is taken for the case of unconditional no branch (NLBR).

5. The short-branch instructions are two bytes long. The first byte specifies the condition to be tested, and the second specifies the branching address.

   The short branch instruction can:
   a. Branch unconditionally
   b. Test for D = 0 or D ≠ 0
   c. Test for DF = 0 or DF = 1
   d. Test for Q = 0 or Q = 1
   e. Test the status (1 or 0) of the four EF flags
   f. Effect an unconditional no branch

   If the tested condition is met, then branching takes place; the branching address byte is loaded into the low-order byte position of the current program counter. This effects a branch within the current 256-byte page of the memory, i.e., the page which holds the branching address. If the tested condition is not met, the branching address byte is skipped over, and the next instruction in sequence is fetched and executed. This same action is taken in the case of unconditional no branch (NBR).

6. The skip instructions are one byte long. There is one Unconditional Short-Skip (SKP) and eight Long-Skip instructions.

   The Unconditional Short-Skip instruction takes 2 cycles to complete (1 fetch + 1 execute). Its action is to skip over the byte following it. Then the next instruction in sequence is fetched and executed. This SKP instruction is identical to the unconditional no-branch instruction (NBR) except that the skipped-over byte is not considered part of the program.
   The Long-Skip instructions take three cycles to complete (1 fetch + 2 execute).

   They can:
   a. Skip unconditionally
   b. Test for D = 0 or D ≠ 0
   c. Test for DF = 0 or DF = 1
   d. Test for Q = 0 or Q = 1
   e. Test for IE = 1

   If the tested condition is met, then Long Skip takes place; the current program counter is incremented twice. Thus two bytes are skipped over, and the next instruction in sequence is fetched and executed. If the tested condition is not met, then no action is taken. Execution is continued by fetching the next instruction in sequence.

TABLE 2. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES

STATE	I	N	SYMBOL	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{\text{MRD}}$	$\overline{\text{MWR}}$	N LINES	NOTES
S1	RESET			0 → I, N, Q, X, P; 1 → IE	00	XXXX	1	1	0	1
	Initialize, Not Programmer Accessible			0000 → R	00	XXXX	1	1	0	2
S0		FETCH		MRP → I, N; RP + 1 → RP	MRP	RP	0	1	0	3
S1	0	0	IDL	IDLE	MR0	RO	0	1	0	4, Fig. 8
	0	1 - F	LDN	MRN → D	MRN	RN	0	1	0	Fig. 8
	1	0 - F	INC	RN + 1 → RN	Float	RN	1	1	0	Fig. 6
	2	0 - F	DEC	RN - 1 → RN	Float	RN	1	1	0	Fig. 6
	3	0 - F	Short Branch	Taken: MRP → RP.0 Not Taken; RP + 1 → RP	MRP	RP	0	1	0	Fig. 8
	4	0 - F	LDA	MRN → D; RN + 1 → RN	MRN	RN	0	1	0	Fig. 8
	5	0 - F	STR	D → MRN	D	RN	1	0	0	Fig. 7
	6	0	IRX	RX + 1 → RX	MRX	RX	0	1	0	Fig. 7
	6	1	OUT 1	MRX → BUS; RX + 1 → RX	MRX	RX	0	1	1	Fig. 11
		2	OUT 2						2	Fig. 11
		3	OUT 3						3	Fig. 11
		4	OUT 4						4	Fig. 11
		5	OUT 5						5	Fig. 11
		6	OUT 6						6	Fig. 11
		7	OUT 7						7	Fig. 11
		9	INP 1	BUS → MRX, D	Data from I/O Device	RX	1	0	1	Fig. 10
		A	INP 2						2	Fig. 10
		B	INP 3						3	Fig. 10
		C	INP 4						4	Fig. 10
		D	INP5						5	Fig. 10
		E	INP6						6	Fig. 10
		F	INP7						7	Fig. 10
	7	0	RET	MRX → (X, P); RX + 1 → RX; 1 → IE	MRX	RX	0	1	0	Fig. 8
		1	DIS	MRX → (X, P); RX + 1 → RX; 0 → IE	MRX	RX	0	1	0	Fig. 8
		2	LDXA	MRX → D; RX + 1 → RX	MRX	RX	0	1	0	Fig. 8
		3	STXD	D → MRX; RX - 1 → RX	D	RX	1	0	0	Fig. 7
		4	ADC	MRX + D + DF → DF, D	MRX	RX	0	1	0	Fig. 8
		5	SDB	MRX - D - DFN → DF, D	MRX	RX	0	1	0	Fig. 8
		6	SHRC	LSB(D) → DF; DF → MSB(D)	Float	RX	1	1	0	Fig. 6
		7	SMB	D - MRX - DFN → DF, D	MRX	RX	0	1	0	Fig. 8
		8	SAV	T → MRX	T	RX	1	0	0	Fig. 7

TABLE 2. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES (Continued)

STATE	I	N	SYMBOL	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{\text{MRD}}$	$\overline{\text{MWR}}$	N LINES	NOTES
S1	7	9	MARK	(X, P) → T, MR2; P → X; R2 - 1 → R2	T	R2	1	0	0	Fig. 7
		A	REQ	0 → Q	Float	RP	1	1	0	Fig. 6
		B	SEQ	1 → Q	Float	RP	1	1	0	Fig. 6
		C	ADCI	MRP + D + DF → DF, D; RP + 1	MRP	RP	0	1	0	Fig. 8
		D	SDBI	MRP - D - DFN → DF, D; RP + 1	MRP	RP	0	1	0	Fig. 8
		E	SHLC	MSB(D) → DF; DF → LSB(D)	Float	RP	1	1	0	Fig. 6
		F	SMBI	D - MRP - DFN → DF, D; RP + 1	MRP	RP	0	1	0	Fig. 8
	8	0 - F	GLO	RN.0 → D	RN.0	RN	1	1	0	Fig. 6
	9	0 - F	GHI	RN.1 → D	RN.1	RN	1	1	0	Fig. 6
	A	0 - F	PLO	D → RN.0	D	RN	1	1	0	Fig. 6
	B	0 - F	PHI	D → RN.1	D	RN	1	1	0	Fig. 6
S1#1	C	0 - 3, 8 - B	Long Branch	Taken: MRP → B; RP + 1 → RP	MRP	RP	0	1	0	Fig. 9
#2				Taken: B → RP.1; MRP → RP.0	M(RP + 1)	RP + 1	0	1	0	Fig. 9
S1#1				Not Taken: RP + 1 → RP	MRP	RP	0	1	0	Fig. 9
#2				Not Taken: RP + 1 → RP	M(RP + 1)	RP + 1	0	1	0	Fig. 9
S1#1		5 6 7 C D E F	Long Skip	Taken: RP + 1 → RP	MRP	RP	0	1	0	Fig. 9
#2				Taken: RP + 1 → RP	M(RP + 1)	RP + 1	0	1	0	Fig. 9
S1#1				Not Taken: No Operation	MRP	RP	0	1	0	Fig. 9
#2				Not Taken: No Operation	MRP	RP	0	1	0	Fig. 9
S1#1		4	NOP	No Operation	MRP	RP	0	1	0	Fig. 9
#2				No Operation	MRP	RP	0	1	0	Fig. 9
S1	D	0 - F	SEP	N → P	NN	RN	1	1	0	Fig. 6
	E	0 - F	SEX	N → X	NN	RN	1	1	0	Fig. 6
S1	F	0	LDX	MRX → D	MRX	RX	0	1	0	Fig. 8
		1 2 3 4 5 7	OR AND XOR ADD SD SM	MRX OR D → D MRX AND D → D MRX XOR D → D MRX + D → DF, D MRX - D → DF, D D - MRX → DF, D	MRX	RX	0	1	0	Fig. 8
		6	SHR	LSB(D) → DF; 0 → MSB(D)	Float	RX	1	1	0	Fig. 6

TABLE 2. CONDITIONS ON DATA BUS AND MEMORY ADDRESS LINES DURING ALL MACHINE STATES (Continued)

STATE	I	N	SYMBOL	OPERATION	DATA BUS	MEMORY ADDRESS	$\overline{MRD}$	$\overline{MWR}$	N LINES	NOTES
S1	F	8	LDI	MRP → D; RP + 1 → RP	MRP	RP	0	1	0	Fig. 8
		9	ORI	MRP OR D → D; RP + 1 → RP						
		A	ANI	MRP AND D → D; RP + 1 → RP						
		B	XRI	MRP XOR D → D; RP + 1 → RP						
		C	ADI	MRP + D → DF, D; RP + 1 → RP						
		D	SDI	MRP - D → DF, D; RP + 1 → RP						
		F	SMI	D - MRP → DF, D; RP +1 → RP						
		E	SHL	MSB(D) → DF; 0 → LSB(D)	Float	RP	1	1	0	Fig. 6
S2	DMA IN			BUS → MR0; R0 + 1 → R0	Data from I/O Device	R0	1	0	0	6, Fig. 12
	DMAOUT			MR0 → BUS; R0 + 1 → R0	MR0	R0	0	1	0	6, Fig. 13
S3	INTERRUPT			X, P → T; 0 → IE, 1 → P; 2 → X	Float	RN	1	1	0	Fig. 14
S1	LOAD			IDLE ($\overline{CLEAR}$, $\overline{WAIT}$ = 0)	M(R0 - 1)	R0 - 1	0	1	0	5, Fig. 8

NOTES:

1. IE = 1, TPA, TPB suppressed, state = S1.
2. BUS = 0 for entire cycle.
3. Next state always S1.
4. Wait for DMA or INTERRUPT.
5. Suppress TPA, wait for DMA.
6. IN REQUEST has priority over OUT REQUEST.
7. See Timing Waveforms, Figure 5 through Figure 14 for machine cycles.

Operating and Handling Considerations

Handling

All inputs and outputs of Intersil CMOS devices have a network for electrostatic protection during handling.

Operating

Operating Voltage - During operation near the maximum supply voltage limit care should be taken to avoid or suppress power supply turn-on and turn-off transients, power supply ripple, or ground noise; any of these conditions must not cause V_{DD} - V_{SS} to exceed the absolute maximum rating.

Input Signals - To prevent damage to the input protection circuit, input signals should never be greater than V_{DD} nor less than V_{SS}. Input currents must not exceed 10mA even when the power supply is off.

Unused Inputs - A connection must be provided at every input terminal. All unused input terminals must be connected to either V_{DD} or V_{SS}, whichever is appropriate.

Output Short Circuits - Shorting of outputs to V_{DD} or V_{SS} may damage CMOS devices by exceeding the maximum device dissipation.

All Intersil U.S. products are manufactured, assembled and tested utilizing ISO9000 quality systems.
Intersil Corporation’s quality certifications can be viewed at www.intersil.com/design/quality

Intersil products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design, software and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.

For information regarding Intersil Corporation and its products, see www.intersil.com

CDP1802AC/3

March 1997

High-Reliability CMOS 8-Bit Microprocessor

Features

- **For Use In Aerospace, Military, and Critical Industrial Equipment**
- **Minimum Instruction Fetch-Execute Time of 4.5µs (Maximum Clock Frequency of 3.6MHz) at V_{DD} = 5V, T_A = +25°C**
- **Operation Over the Full Military Temperature Range . . . . -55°C to +125°C**
- **Any Combination of Standard RAM and ROM Up to 65,536 Bytes**
- **8–Bit Parallel Organization With Bidirectional Data Bus and Multiplexed Address Bus**
- **16 x 16 Matrix of Registers for Use as Multiple Program Counters, Data Pointers, or Data Registers**
- **On-Chip DMA, Interrupt, and Flag Inputs**
- **High Noise Immunity . . . . 30% of V_{DD}**

Ordering Information

PACKAGE	TEMP. RANGE (°C)	5V - 3.2MHz	PKG NO.
SBDIP	-55 to 125	CDP1802ACD3	D40.6

Description

The CDP1802A/3 High-Reliability LSI CMOS 8-bit register oriented Central-Processing Unit (CPU) is designed for use as a general purpose computing or control element in a wide range of stored-program systems or products.

The CDP1802A/3 includes all of the circuits required for fetching, interpreting, and executing instructions which have been stored in standard types of memories. Extensive input/output (I/O) control features are also provided to facilitate system design.

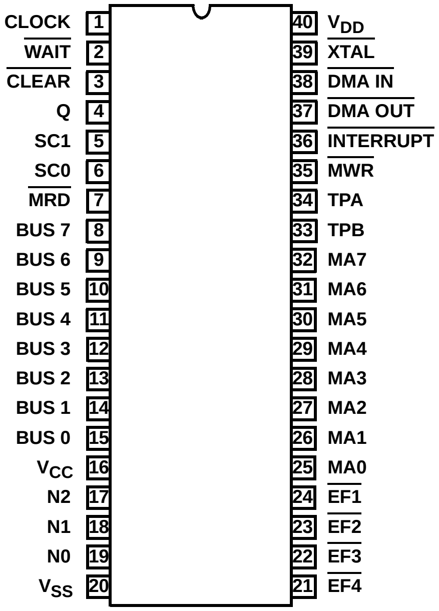
The 1800 Series Architecture is designed with emphasis on the total microcomputer system as an integral entity so that systems having maximum flexibility and minimum cost can be realized. The 1800 Series CPU also provides a synchronous interface to memories and external controllers for I/O devices, and minimizes the cost of interface controllers. Further, the I/O interface is capable of supporting devices operating in polled, interrupt-driven, or direct memory-access modes.

The CDP1802AC/3 is functionally identical to its predecessor, the CDP1802. The "A" version includes some performance enhancements and can be used as a direct replacement in systems using the CDP1802.

This type is supplied in 40 lead dual-in-line sidebrazed ceramic packages (D suffix).

Pinout

CDP1802AC/3 (SBDIP)
TOP VIEW



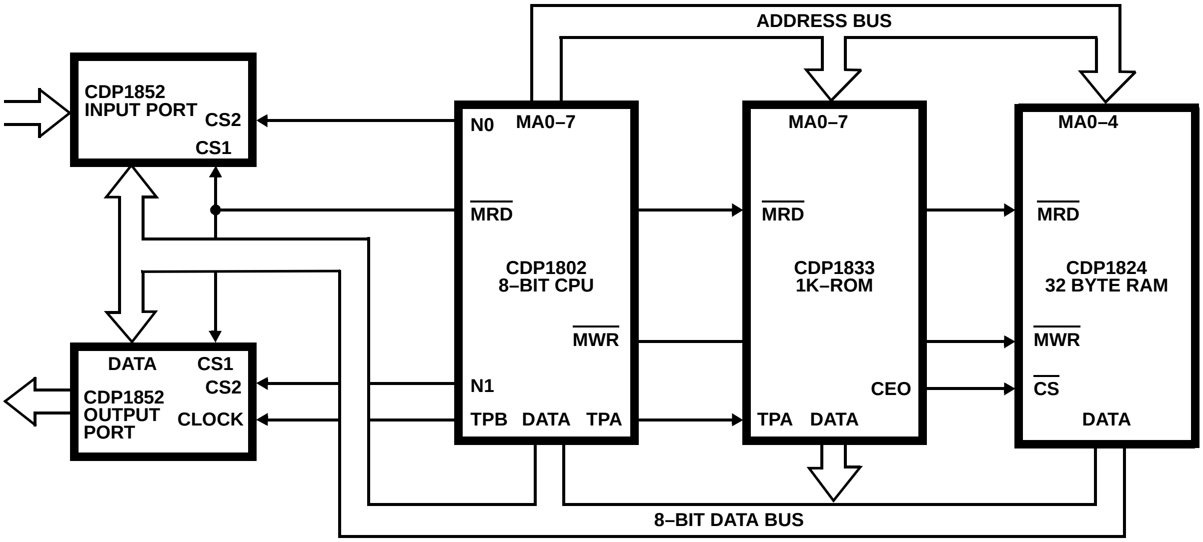


FIGURE 1. TYPICAL CDP1802A/3 SMALL MICROPROCESSOR SYSTEM

Absolute Maximum Ratings

DC Supply Voltage Range, (V_{DD})
(All Voltages Referenced to V_{SS} Terminal)
CDP1802AC/3 . . . . -0.5V to +7V
Input Voltage Range, All Inputs . . . . -0.5V to V_{DD} +0.5V
DC Input Current, any One Input. . . . . ±10mA

Thermal Information

Thermal Resistance (Typical)	θ_{JA} (^{O}C/W)	θ_{JC} (^{O}C/W)
SBDIP Package	55	15

Device Dissipation Per Output Transistor
T_A = Full Package Temperature Range 100mW
Operating Temperature Range (T_A)
Package Type D . . . .-55^{O}C to +125^{O}C
Storage Temperature Range (T_{STG}). . . . .-65^{O}C to +150^{O}C
Lead Temperature (During Soldering)
At distance 1/16 ± 1/32 In. (1.59 ±0.79mm)
from case for 10s max . . . . +265^{O}C

CAUTION: Stresses above those listed in “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

Recommended Operating Conditions

T_A = Full Package Temperature Range. For maximum reliability, operating conditions should be selected so that operation is always within the following ranges

PARAMETER	MIN	MAX	UNITS
DC Operating Voltage Range	4	6.5	V
Input Voltage Range	V_{SS}	V_{DD}	V
Maximum Clock Input Rise or Fall Time	-	1	µs

Performance Specifications

PARAMETER	V_{DD} (V)	-55^{O}C TO +25^{O}C	+125^{O}C	UNITS
Minimum Instruction Time (Note 1)	5	4.5	5.9	µs
Maximum DMA Transfer Rate	5	450	340	Kbytes/s
Maximum Clock Input Frequency, Load Capacitance (C_L) = 50pF, f_{CL}	5	DC-3.6	DC-2.7	MHz

NOTE:

1. Equals 2 machine cycles - one Fetch and one Execute operation for all instructions except Long Branch and Long Skip, which require 3 machine cycles - one Fetch and two Execute operations.

Static Electrical Specifications

All Limits are 100% Tested

	CONDITIONS			-55^{O}C, +25^{O}C		+125^{O}C		
PARAMETER	V_{OUT} (V)	V_{IN}, (V)	V_{CC}, V_{DD} (V)	MIN	MAX	MIN	MAX	UNITS
Quiescent Device Current, I_{DD}	-	-	5	-	100	-	250	µA
Output Low Drive (Sink) Current (Except $\overline{XTAL}$), I_{OL}	0.4	0, 5	5	1.20	-	0.90	-	mA
$\overline{XTAL}$	0.4	5	5	185	-	140	-	µA
Output High Drive (Source) Current (Except $\overline{XTAL}$), I_{OH}	4.6	0, 5	5	-	-0.30	-	-0.20	mA
$\overline{XTAL}$	4.6	0	5	-	-135	-	-100	µA
Output Voltage Low-Level, V_{OL}	-	0, 5	5	-	0.1	-	0.2	V
Output Voltage High-Level, V_{OH}	-	0, 5	5	4.9	-	4.8	-	V

Static Electrical Specifications All Limits are 100% Tested **(Continued)**

PARAMETER	CONDITIONS			-55°C, +25°C		+125°C		UNITS
	V_{OUT} (V)	V_{IN}, (V)	V_{CC}, V_{DD} (V)	MIN	MAX	MIN	MAX	
Input Low Voltage, V_{IL}	0.5, 4.5	-	5	-	1.5	-	1.5	V
Input High Voltage, V_{IH}	0.5, 4.5	-	5	3.5	-	3.5	-	V
Input Leakage Current, I_{IN}	Any Input	0, 5	5	-	±1	-	±5	µA
Three-State Output Leakage Current, I_{OUT}	0, 5	0, 5	5	-	±1	-	±5	µA

NOTE:

2. 5V level characteristics apply to Part No. CDP1802AC/3, and 5V and 10V level characteristics apply to part No. CDP1802A/3.

Timing Specifications As a Function of T (T = 1/fCLOCK), C_L = 50 pF

PARAMETER	V_{DD} (V)	**LIMITS** (NOTE 3)		UNITS
		-55°C, +25°C	+125°C	
High-Order Memory-Address Byte Setup to TPA ↘ Time, t_{SU}	5	2T-450	2T-580	ns
High-Order Memory-Address Byte Hold After TPA Time, t_H	5	T/2 +0	T/2 +0	ns
Low-Order Memory-Address Byte Hold After WR Time, t_H	5	T-30	T-40	ns
CPU Data to Bus Hold After WR Time, t_H	5	T-170	T-250	ns
Required Memory Access Time Address to Data, t_{ACC}	5	5T-300	5T-400	ns

NOTE:

3. These limits are not directly tested.

Implicit Specifications (Note 4) T_A = -55°C to +25°C

PARAMETER		SYMBOL	V_{DD} (V)	TYPICAL VALUES	UNITS
Typical Total Power Dissipation Idle “00” at M(0000), C_L = 50pF	f = 2MHz	-	5	4	mW
Effective Input Capacitance any Input	-	C_{IN}	-	5	pF
Effective Three-State Terminal Capacitance Data Bus	-		-	7.5	pF
Minimum Data Retention Voltage	-	V_{DR}	-	2.4	V
Data Retention Current	-	I_{DR}	2.4	10	µA

NOTE:

4. These specifications are not tested. Typical values are provided for guidance only.

Dynamic Electrical Specifications C_L = 50pF, Timing Measurement at 0.5 V_{DD} Point

PARAMETERS	V_{DD} (V)	-55°C TO +25°C		+125°C		UNITS
		MIN	MAX	MIN	MAX	
Progagation Delay Times, t_{PLH}, t_{PHL}						
Clock to TPA, TPB	5	-	275	-	370	ns
Clock-to-Memory High Address Byte, t_{PLH}, t_{PHL}	5	-	725	-	950	ns
Clock-to-Memory Low Address Byte Valid, t_{PLH}, t_{PHL}	5	-	340	-	425	ns
Clock to $\overline{MRD}$, t_{PLH}, t_{PHL}	5	-	340	-	425	ns
Clock to $\overline{MWR}$, t_{PLH}, t_{PHL}	5	-	275	-	370	ns
Clock to (CPU DATA to BUS) Valid, t_{PLH}, t_{PHL}	5	-	430	-	550	ns
Clock to State Code, t_{PLH}, t_{PHL}	5	-	440	-	550	ns
Clock to Q, t_{PLH}, t_{PHL}	5	-	375	-	475	ns
Clock to N (0 - 2), t_{PLH}, t_{PHL}	5	-	400	-	525	ns
Interface Timing Requirements (Note 5)						
Data Bus Input Setup, t_{SU}	5	10	-	10	-	ns
Data Bus Input Hold, t_H	5	175	-	230	-	ns
$\overline{DMA}$ Setup, t_{SU}	5	10	-	10	-	ns
$\overline{DMA}$ Hold, t_H	5	200	-	270	-	ns
Interrupt Setup, t_{SU}	5	10	-	10	-	ns
Interrupt Hold, t_H	5	175	-	230	-	ns
$\overline{WAIT}$ Setup, t_{SU}	5	30	-	30	-	ns
$\overline{EF1-4}$ Setup, t_{SU}	5	20	-	20	-	ns
$\overline{EF1-4}$ Hold, t_H	5	100	-	135	-	ns
Required Pulse Width Times						
$\overline{CLEAR}$ Pulse Width, t_{WL}	5	150	-	200	-	ns
$\overline{CLOCK}$ Pulse Width, t_{WL}	5	140	-	185	-	ns

NOTE:

5. Minimum input setup and hold times required by Part CDP1802AC/3.

Performance Curves

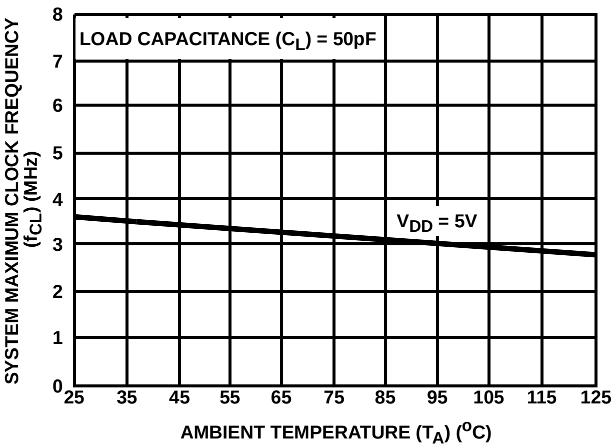


FIGURE 2. TYPICAL MAXIMUM CLOCK FREQUENCY AS A FUNCTION OF TEMPERATURE

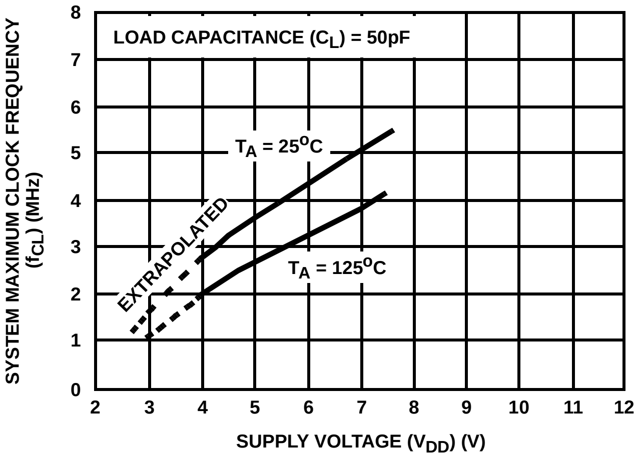


FIGURE 3. TYPICAL MAXIMUM CLOCK FREQUENCY AS A FUNCTION OF SUPPLY VOLTAGE

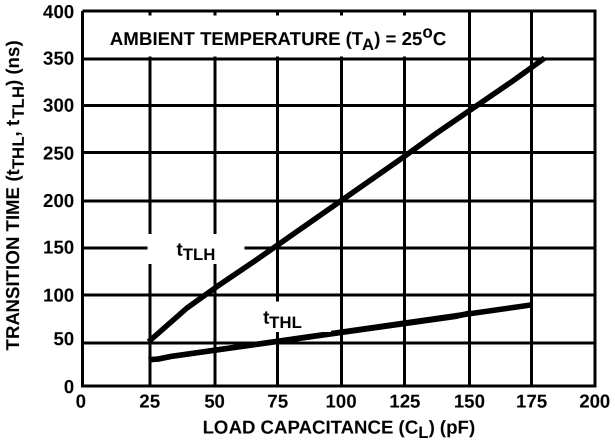


FIGURE 4. TYPICAL TRANSITION TIME vs LOAD CAPACITANCE

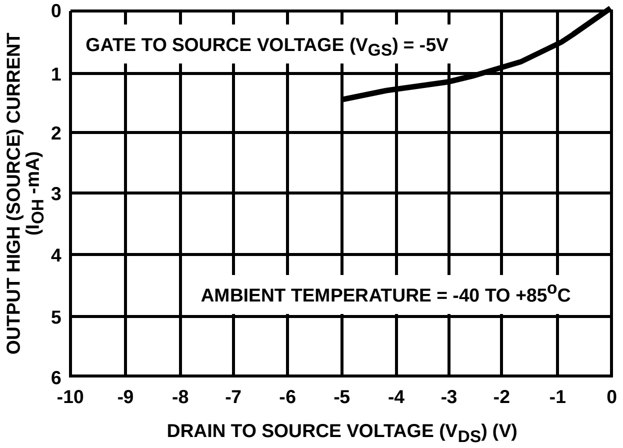
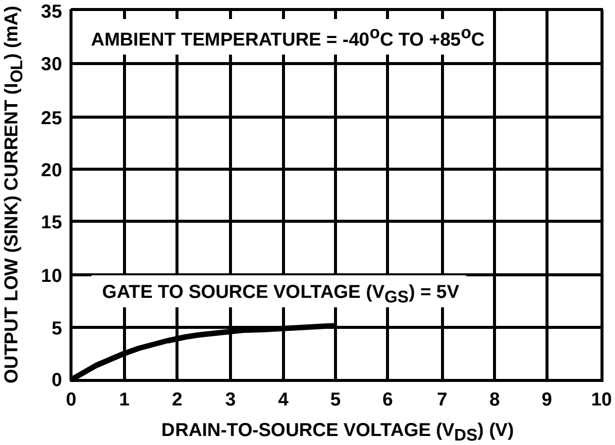
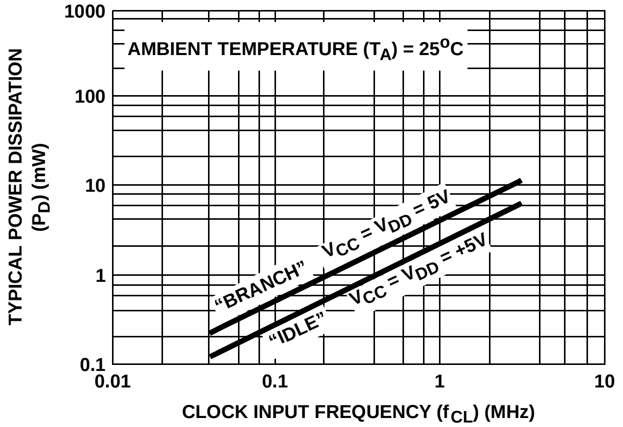


FIGURE 5. MINIMUM OUTPUT HIGH (SOURCE) CURRENT CHARACTERISTICS





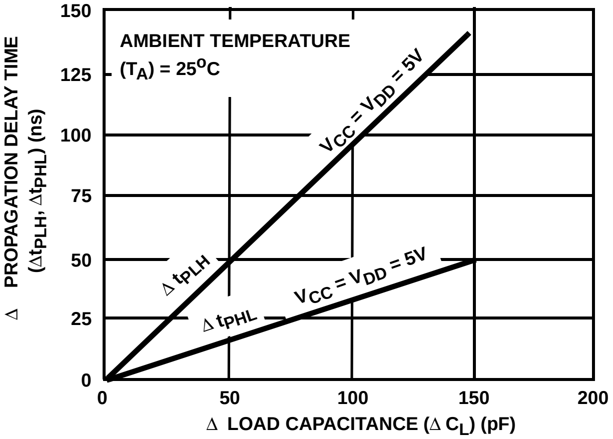
NOTES:

6. Idle = "00" at M (0000)
7. Branch = "3707" at M (8107)

FIGURE 6. MINIMUM OUTPUT LOW (SINK) CURRENT CHARACTERISTICS

FIGURE 7. TYPICAL POWER DISSIPATION AS A FUNCTION OF CLOCK FREQUENCY FOR BRANCH INSTRUCTION AND IDLE INSTRUCTION

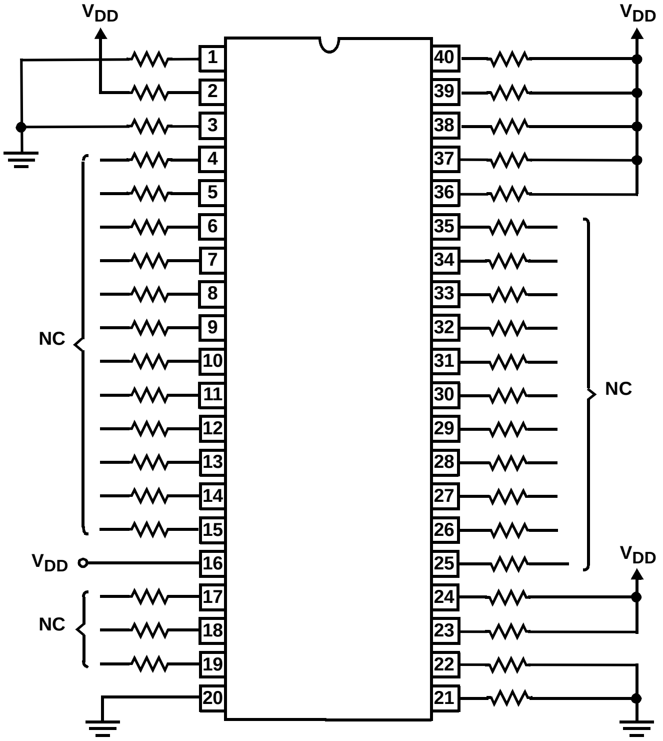
Performance Curves (Continued)



NOTE: Any output except $\overline{\text{XTAL}}$.

FIGURE 8. TYPICAL CHANGE IN PROPAGATION DELAY AS A FUNCTION OF A CHANGE IN LOAD CAPACITANCE

Burn-In Circuit



ALL RESISTORS ARE 47kΩ ±20%

TYPE	V_{DD}	TEMPERATURE	TIME
CDP1802AC	7V	+125°C	160 Hours

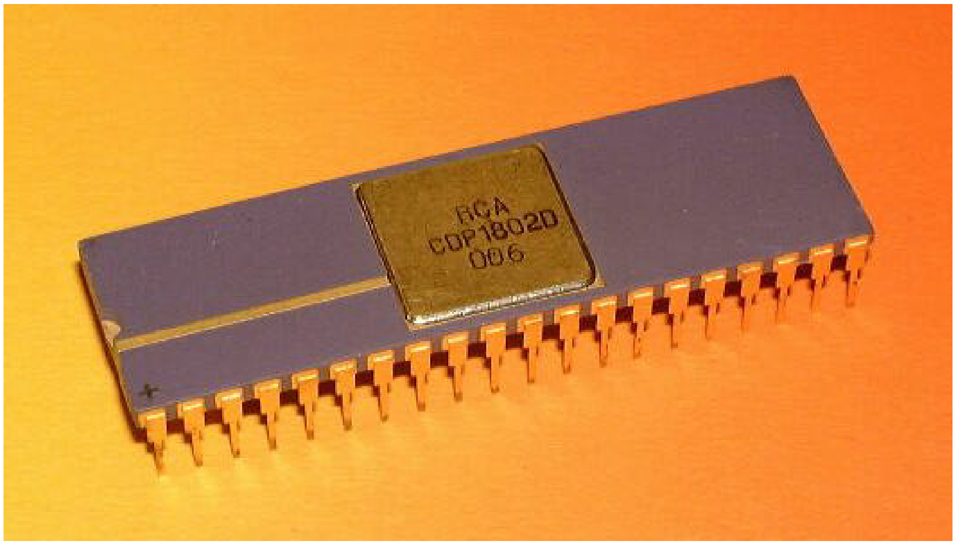
FIGURE 9. BIAS/STATIC BURN-IN CIRCUIT

All Intersil U.S. products are manufactured, assembled and tested utilizing ISO9000 quality systems.
Intersil Corporation's quality certifications can be viewed at www.intersil.com/design/quality

Intersil products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design, software and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.

For information regarding Intersil Corporation and its products, see www.intersil.com

RCA 1802 Microprocessor



Prepared By:
Ben Knox
Henna Karim
Micah Sumner
Morgan Grinstead
John Redmon

Prepared For:
Dr. Joe Dumas
College of Engineering and Computer Science
University of Tennessee at Chattanooga
November 13, 2002
Computer Science 460 Section 001

Table of Contents

Introduction

The RCA 1802 was designed and introduced in 1974. The chip was capable of running at a clock frequency of 6.4 MHz using a +10V power supply. The processor was designed and fabricated in CMOS. The chip utilized 16 16-bit registers that could be accessed as 32 8-bit registers [10]. It was an 8 bit processor with 16 bit addressing. It was extremely simple and had the flexibility of a large register set. The 1802 had a single accumulator register and a 16x16 register file [10]. This versatile chip was the first microprocessor used in space, and RCA produced videogames based on it. The 1802 reached its peak of popularity in 1977, and at that time sold for a little less than $30.00 as stated by Popular Mechanics [11]. The last commercial microcomputer produced with the 1802 was a 1983 machine named the Comx 35. By 1983 the 1802 was no longer cutting edge technology and the slow speed of the machine made it less efficient. We will now look the 1802 microprocessor in more detail, including its inner workings such as registers and instructions.

Architectural Classification

The RCA 1802 has an extremely simple design. It is a textbook example of the Von-Neumann architecture. It has a single bus for instructions and data, and addresses a single memory. The processor does not have temporal or spatial parallelism. It performs an instruction to completion before it fetches and executes the next instruction. [1]

The 1802 has many characteristics that would later be implemented in RISC processors. It is not, however, a RISC chip. The CPU has a relatively large set of 16 general-purpose 16-bit registers. The instruction set, with only 91 instructions, is very simple. Despite these traits the 1802 cannot be classified as a RISC processor for several reasons. The load-store trait of RISC is not present. Many of the instructions, such as ADI, access memory directly. [1] The instruction size is not the same for all instructions. Most instructions are one byte, but any instruction that includes data is two bytes, and any instruction that contains an address is three bytes. The 1802 is unable to perform any instructions in a single clock cycle. In fact, most instructions took two machine cycles of eight clock pulses to perform their intended task, and in some cases the instruction took three machine cycles. [6]

Intended Applications

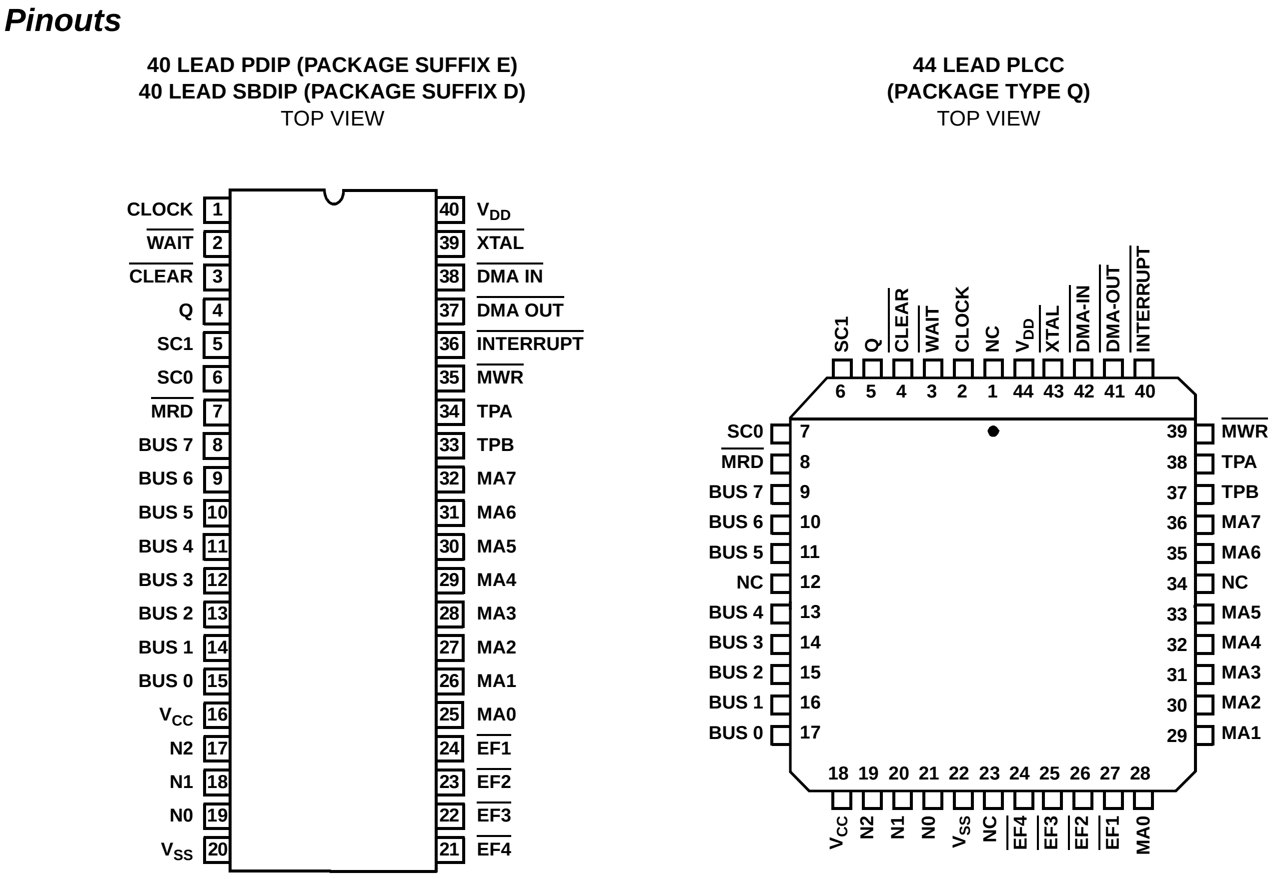
The 1802 was designed for use as general-purpose computing or control elements in a wide range of stored program systems or products. It was designed with emphasis on maximum flexibility and minimum cost, thus this chip was used in a wide range of devices.

The 1802 was the first microprocessor in space via the Defense Meteorological Satellite Program 5D-1 spacecraft launched in 1976. It was also included on the following spacecrafts: OSCAR satellite, U_oSAT-1, U_oSAT-2, Voyager, Viking, and Galileo. [1] One reason for its space use was that the 1802 was fabricated on sapphire, which leads to radiation and static resistance, ideal for space travel. [12]

This versatile chip was also used in Chrysler electronic ignitions, RCA and Radio Shack video games, RCA video terminals, and ETI-660 computers. [14] Swiss payphone manufacturer Sodeco-Sia used it for phones in France, Austria, and third world countries where its low power allowed the unit to work entirely from the power of the phone line. [13]

In August & September of 1976 and also March & July of 1977, Popular Mechanics ran a series of articles written by Joseph Weisbeckes on how to build a hobbyist computer. Mr. Weisbeckes called this computer the COSMAC ELF. [17]

Figure 1: 1802 Pin Layout



Due to its age, any specific information concerning the inner workings of the RCA 1802's control unit is very hard to find. All the information accumulated points to a simple, non-pipelined, hardwired control unit.

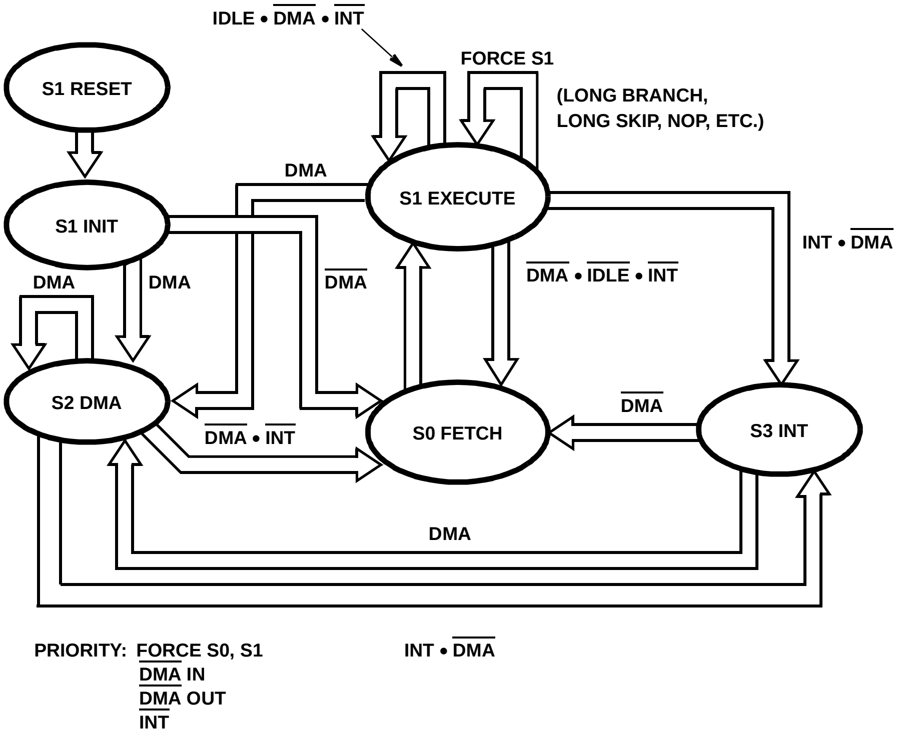
A single machine cycle consists of eight pulses of the clock. The following is a discussion of the communication lines entering and leaving the control unit. [1]

The control unit has two state code lines, SC0 and SC1, which provide information on its current activity. Figure 2 provides a state table of their outputs and what they represent. The control unit generally alternates between the S0, an instruction fetch, and S1, instruction execution, states. It takes one complete machine cycle to fetch or execute an instruction. The only exceptions are the long-branch and the long-skip instructions. These instructions require a single machine cycle to be fetched and two machine cycles to be executed. Figure 3 provides a state transition diagram for the states of the control unit. The minimum instruction time for 2 machine cycle instruction executed on the CDP1802 running at 6.4 MHz and +10v is 2.5 microseconds. [1]

Figure 2: State Table

STATE TYPE	STATE CODE LINES	
	SC1	**SC0**
S0 (Fetch)	L	L
S1 (Execute)	L	H
S2 (DMA)	H	L
S3 (Interrupt)	H	H

Figure 3: State Transition Diagram



The control unit has a write pulse, ~MWR, and read level, ~MRD, output. (The tilde represents NOT.) A negative pulse on the ~MWR represents a write cycle. A negative pulse on the ~MRD represents a read cycle. [1]

The TPA and TPB, timing pulses, are sent out from the control unit. These are the positive pulses that are sent out once in a machine cycle. They are used by the I/O controllers to synchronize interaction with the data bus. The trailing edge of the TPA signals the memory system to latch the high bits on the address bus. The low order bits are placed on the bus at the termination of the TPA. [1]

The Q flip-flop is a single bit output from the control unit that can be set or reset by a program. It can then be checked to see if a conditional branch is to be taken by the system. [1]

The ~XTAL output is to be used in conjunction with an external clock crystal if the on board oscillator is used. It is used with the clock input terminal. [1]

There are two control lines, ~CLEAR and ~WAIT, that lead into the control unit. Figure 4 provides a state table for these inputs and the functions they execute. [1]

$\overline{\text{CLEAR}}$	**$\overline{\text{WAIT}}$**	**MODE**
L	L	LOAD
L	H	RESET
H	L	PAUSE
H	H	RUN

Figure 4: Control Line State Table

There are three input lines that lead into the control unit that concern I/O requests: ~DMA IN, ~DMA OUT, ~INT. The ~INT line signals an interrupt. The ~DMA IN and ~DMA OUT lines signal that a DMA action is requested. [1]

There are four I/O flags attached to the control unit, ~EF1 - ~EF4. These flags are used by the I/O controllers to transfer status information to the control unit. They can also be used along with the interrupt line to establish a priority system among the interrupts. They can also be used by the I/O devices to alert the control unit that they need attention. The flags are sampled at the beginning of the S1 cycle. [1]

The N and I registers are connected to the instruction decode logic of the control unit. [1]

CPU Registers

Registers

The key feature of the RCA 1802 was its large register set. The 1802 had 16 sixteen-bit general purpose registers. Figure 5 shows the general organization of the registers in the CPU. The following is a description of the registers and their purpose.

R(0x0) – R(0xF)

The general purpose registers are all 16-bit registers. R(3) designates the register selected by the binary code 0011. They can also be accessed as 32 eight-bit registers. R(3).0 refers to the low order byte of R(3). R(3).1 refers to the high order byte of R(3). These registers have a wide variety of uses. One of them will be pointed to by the P register to become the program counter. Another may be pointed to by the X register to be a data pointer. In each case the register will hold the address of a location in memory. The N register may also point to one of these registers to be used in an instruction. The registers can also hold data that can be sent along to data bus to the D register to be used by the ALU. One of the registers can also be used as a counter by being sent through the incrementer attached to the A register. The 1802 does not have a stack pointer, but clever use of one the general purpose registers can allow the user to implement a stack. [16]

A

The A register is a 16-bit register. It is a temporary storage register that is connected to the general registers and the incrementer/decrementer. It holds data from one of the general registers. It can then send it through the incrementer/decrementer to be operated upon. Then it can either put the data back into a register or send it to the data bus. The A register is instrumental in the implementation of a stack. The address held in the general register that is being used as the stack pointer can be sent to the A register to be decremented for a stack push or incremented for a stack pop. The new address would then be stored back in the original general register. [1]

D

The D register is the accumulator. It is one byte or eight bits in length. The D register holds all data to be manipulated by the ALU [16].

P

The P register is a four-bit register that contains the address of one of the general purpose registers. Four bits are all that is needed to address 16 individual registers or memory addresses. The register pointed to by the P register becomes the program counter. The program counter holds the address of the next instruction to be executed in RAM. [6]

X

The X register is a four-bit register that contains the address of one of the general purpose registers. The register pointed to by the X register becomes a data pointer. The register holds the address of data that can be used in some ALU operations, input operations, output operations, and miscellaneous operations. [1]

T

The T register is an eight-bit register. It is used to temporarily hold the values of the X and P register when an interrupt occurs. If the processor is currently executing an instruction when an interrupt occurs, the instruction completes execution before X and P are loaded into T. After the interrupt is handled, X and P are restored. [1]

I, N

The I and N registers are both four-bits in length. They are used together as a makeshift Instruction Register. The N register holds the low-order bits of the instruction. These bits either point to a general purpose register or act as a special code. The I register holds the high-order bits of the instruction. These bits designate the type of instruction. [16]

DF

The DF register is a single-bit register. Technically it is named the Data Flag, but it operates similarly to a carry flag. It contains the carry out of an ALU operation. [6]

IE

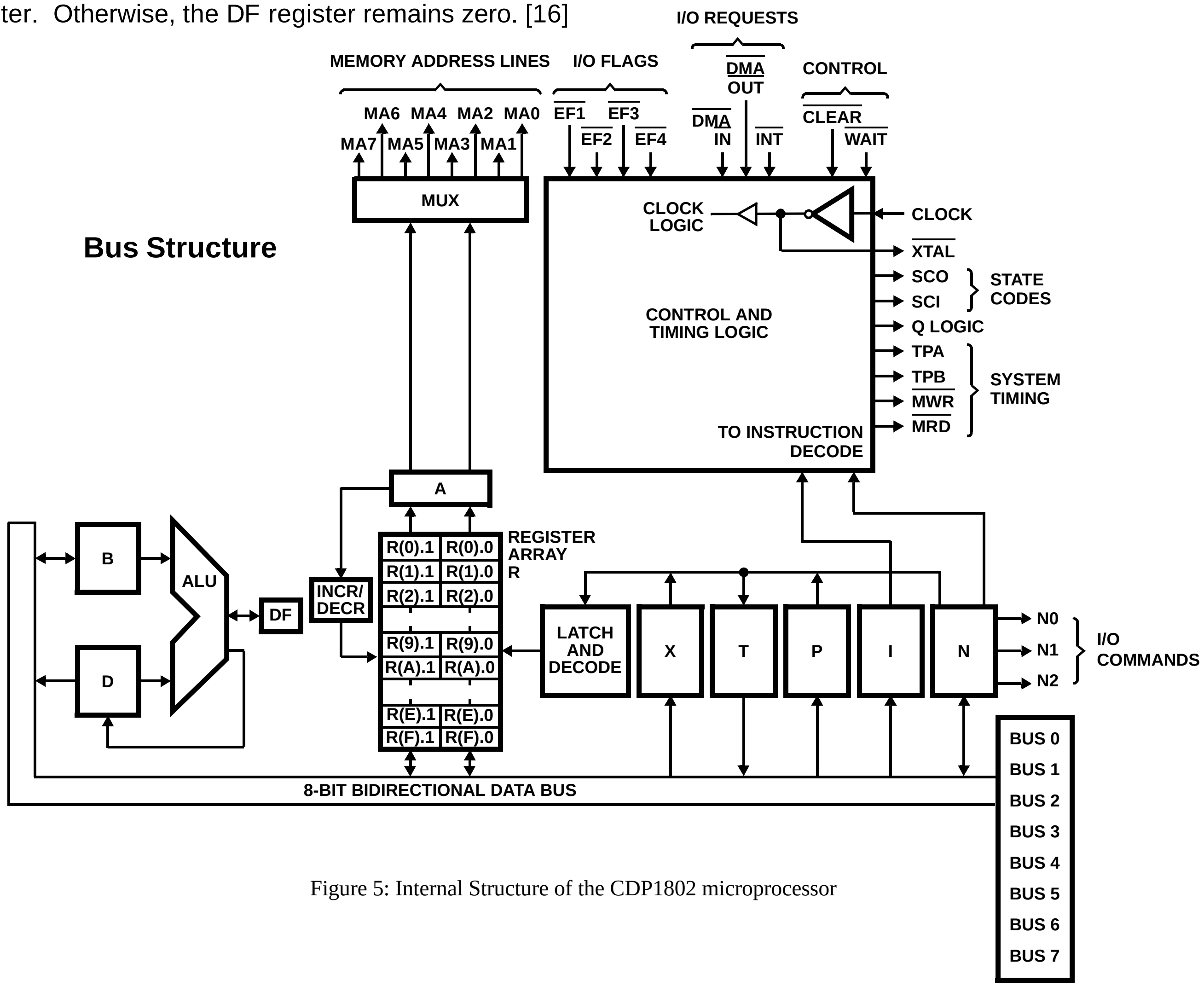
The IE register is a single-bit register. It is the Interrupt Enable flag. If it is one then the processor accepts interrupts. If the flag is zero the processor denies interrupts. [6]

Internal Bus Structure

See Bus Structure.

Functional Unit

The functional unit of the RCA 1802 consists of a single eight-bit ALU. The ALU performs arithmetic and logical operations on data. It receives one operand from the D register and the other operand from the data bus. The result of the operation is placed in the D register. If there is a carry a one is placed in the DF register. Otherwise, the DF register remains zero. [16]



Bus Structure

Figure 5: Internal Structure of the CDP1802 microprocessor

The RCA 1802 Chip used 8-bit parallel organization with bi-directional data bus and multiplexed address bus. The 1800 series CPU utilizes a synchronous interface to memories and external controllers for I/O devices. The I/O interface allowed support of devices operating in polled, interrupt driven, or direct memory access (DMA) modes.

The 1802 used N0 to N2 (pins #'s 17-19) as I/O control lines. [1] These lines can address up to 8

Input/Output devices to I/O or vice versa (flow usually indicated by N3 internally). [16]

The INTERRUPT , DMA-IN , and DMA-OUT are 3 I/O Requests that can be sent on the bus.

The 8-bit data bus serves a dual function. The data bus provides for not only external communication but also internal communication within the processor. As Figure 5 shows, the data bus is directly connected to the ALU and all of the registers except DF, IE, and A. This provides the main source of communication within the processor. Intermittent individual connections connect the control logic to the functional units of the processor.

Note: The following figures are given for operation at +10 V. The maximum data transfer rate for the data bus occurs during DMA transfers. This rate is 800 KB/sec. The bus does include some minimum set up and hold times which are as follows: data bus set up = 40 ns, data bus Input Hold = 100 ns. [1]

Expandability is a feature. For a cost of $20, the 1802 could interface with RCA's 1861 "PIXIE" graphics chip using its interrupt line, and one of the four "external flag" input lines. This graphics capability utilized the DMA feature of the 1802. [13] Attaching cascading CD4515's to the data bus permitted a large number of I/O lines to be handled (well over 128). [17] The potential large number of I/O lines allowed for a wide range of I/O devices including cassette interface, audio circuitry, monitors, keyboards, and even a joystick. [13]

Performing I/O

The RCA 1802 performs I/O through an 8-bit data bus connecting the registers, ALU and external bus. To output a memory address the 4-bit X register is given a value which points to one of the sixteen R registers which contains the memory address. The output instructions are 61-67. The lower three bits of the instruction are used to choose one of seven possible devices. This value is output on the lines N0, N1 and N2. The lines may be connected directly to an external device and output only for the instructions 61, 62 and 64, or they may be decoded to connect to up to seven devices. When the output instruction is executed the data is moved onto the bus from the memory position pointed to by R(X), and R(X) is incremented. The automatic incrementing of R(X) allows sequential data to be output quickly. [1]

The input works in much the same way as the output. The instructions to perform output are 69-6F. The lower three bits of these instructions are output on the N0, N1 and N2 lines in the same way as the output, and like the output they can be decoded or connected directly to external devices. The X register points to the R register which contains the address of where to output, but unlike the output instructions the value in R(X) is not incremented after execution. [1]

There is a simple implementation of DMA on the chip. The 1802 has a DMA IN pin and a DMA OUT pin. The DMA pins act the same as interrupts. When a signal is received on the pin a machine cycle is used by the CPU to perform the input or output. [6] This cannot be defined as a true implementation of DMA because the CPU is involved in the data transfer, but it is referred to as DMA. If more than one signal is received DMA-IN has priority over DMA-OUT which has priority above the Interrupt line. When the DMA-IN signal is receive the data from the bus is moved into the address pointed to by R(0), and R(0) is incremented. When the DMA-Out signal is received by the CPU the byte in memory pointed to my R(0) is moved onto the data bus and R(0) is incremented. [1]

There are several chips in the 1800 series designed to support I/O. The 1852 is an 8-bit input/output port. When the mode pin is set to 0 the port acts as an input. When the mode pin is set to 1 the port acts as an output. [7] The 1861 is a video output chip that uses the DMA OUT and INT lines to output a 64x128 bitmap image to an output screen. [8] The 1871 is a support chip that senses input from a mechanical keyboard and outputs the appropriate code to the bus. [9]

Instruction Set

The 1802 RCA processor has an instruction set consisting of 91 opcodes, which are software compatible with 1801 instructions (59 opcodes). The 91 single-byte commands are grouped into five basic types: register, memory and logic, arithmetic, branch skip and control, and I/O byte transfer instructions. Most instructions require two machine cycles. [5] The only exception to this are the long branch and long skip instructions, which take 3 machine cycles. [1] Each instruction is broken into two 4-bit hex digits, designated so that I is the higher order digit and N is the lower order digit. The I word specifies the instruction type and N word either specifies which register is to be used or acts as a special code. Register operation include instructions that count data between internal registers. Memory refers to the commands that provide directions on how to load or store memory bytes. Branching operations provide conditional and unconditional branch instructions. Arithmetic logic instructions provide the common operations: add, subtract, AND, OR, EX-OR and shift while control and I/O commands that take care of the timing and data operations. [5] The control functions facilitate the program interrupts, operations selection, branch and link operations and control the Q flip flop. [2] The I/O functions handle memory loading and data transfer operations into and out of the 1802. See Appendix A for a complete set of 1802 instructions.

Instruction Fetch and Decode

Each CPU instruction is fetched on the first machine cycle and executed during the second machine cycle, except for long branch and long skip instructions that require the first machine cycle to fetch the instruction and on the 2nd and 3rd cycle fetch the address (execute). During the fetch cycle the 4-bits in P are designated to select one of the 16 bit registers as the current program counter. The selected register contains the address of the memory location to be fetched. When the instructions are read out of memory, the high 4-bits of the instruction are loaded into the register and the low 4-bits of the instruction are put into the N-register. The content of the program counter is automatically incremented by one so that it is now pointing to the next byte in memory. [1]

Interrupt Processing

Interrupt services can originate from either I/O devices or user defined programs. The initial steps take by the processor for each type are the same. First, the interrupt request is sent by the device or program across a shared interrupt request line to the predefined interrupt pin number 36 on the CPU. Second, the X and P registers are saved in the temporary register T. However, the registers are only saved after the current instruction is finished executing. Next, Interrupt Enable is set to 0 to inhibit further interrupts from being processed. It is at this point that the handling of the interrupt differs. If a user program caused the interrupt then the scratch pad register R(1) is set as the program counter, and the address of the user defined instruction is loaded into it. Next, the user routine must save the value of T by using the **sav** instruction. The **sav** instruction saves the value of T to the memory position pointed to by R(X). From this point forward the user program has full control on how the interrupt is to be handled. Once the user routine is done executing then it is responsible for reloading the values of X and P. The user routine restores the values of X and P by using either the **ret** or **dis.** The **ret** and **dis** instruction work in almost the exact same way. First, they access memory at the address pointed to by the R(X) register. Next, using the data that is found at this address, the instructions restore the X and P registers. The instructions then increment the R(X) register so that it points to the next available instruction. Finally, if the **ret** instruction was used then a 1 is placed in the IE register to enable interrupts. If the **dis** instruction was used then a 0 is placed in the IE register to disable interrupts. If a device caused the interrupt then the ~EF1 to ~EF4 flags are used in conjunction with the interrupt pin to determine the priority level of the interrupt. Once the priority level has been determined, the CPU loads the address of the handler into R(1) and executes these instructions. Before the completion of the routine the values of X and P are restored. [1]

Memory System

The memory system of the CDP1802 is divided into three different components, RAM, ROM and an optional PROM. The general layout of a CDP1802 system is given in the following diagram. [1]

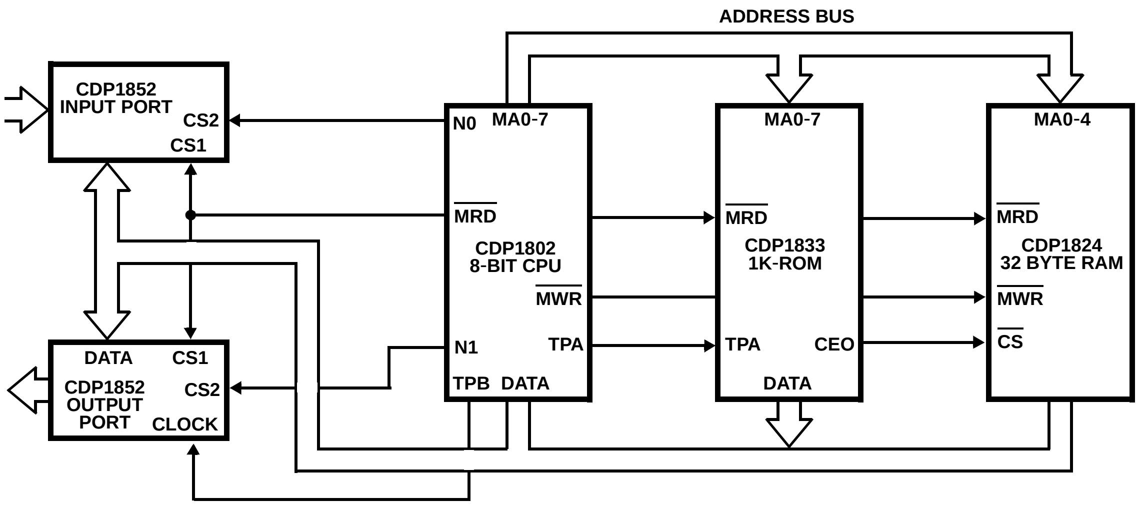


FIGURE 6. TYPICAL CDP1802 SMALL MICROPROCESSOR SYSTEM

As can bee seen from this diagram there exits a single multiplexed address bus that feeds into each memory component. This address bus consists of sub-buses labeled MA0-MA7. There is also a bi-directional data bus that runs from each memory chip to each I/O device and the 8-bit 1802 CPU. [1] In addition, unlike the 1804 and 1806 microprocessors the 1802 does not come bundled with the RAM and ROM on the microprocessor chip. Rather each component must be purchased and installed separately. [1] Since the 1802 decodes the high-order address bits to select between memory chips, it uses a high-order memory interleaving scheme. [1]

RAM

The 1802's RAM takes the form of a CDP1824 chip(s). The 1824 is a 32 word by 8-bit fully static CMOS random access memory. The time it takes to access RAM is about 320ns. There are three signals that are associated with the 1824, ~MRD, ~MWR, and ~CS[1]. The ~CS signal acts as a chip select and is present for memory expansions. The ~MRD and ~MWR signals indicate whether a memory read or write has been requested. [2]

A 1802 memory address consists of 16-bits. During the memory R/W cycle the high order bytes are placed on the address bus first. If there are more than one random access memory chip installed the upper two bits are used for the chip selection, otherwise they are not used. The remaining high-order bits are latched via the TPA clock timing pulse. However, if all the 8 high-order bits are latched then 64K of memory can be obtained. Following the completion of the TPA timing pulse, the lower order bytes appear on the address bus. [1] The lower five low-order address bits are then used as the address within the selected chip. [2]

ROM

ROM can be implemented in a variety of ways. One of the most common ways is indicated in the below figure. [3]

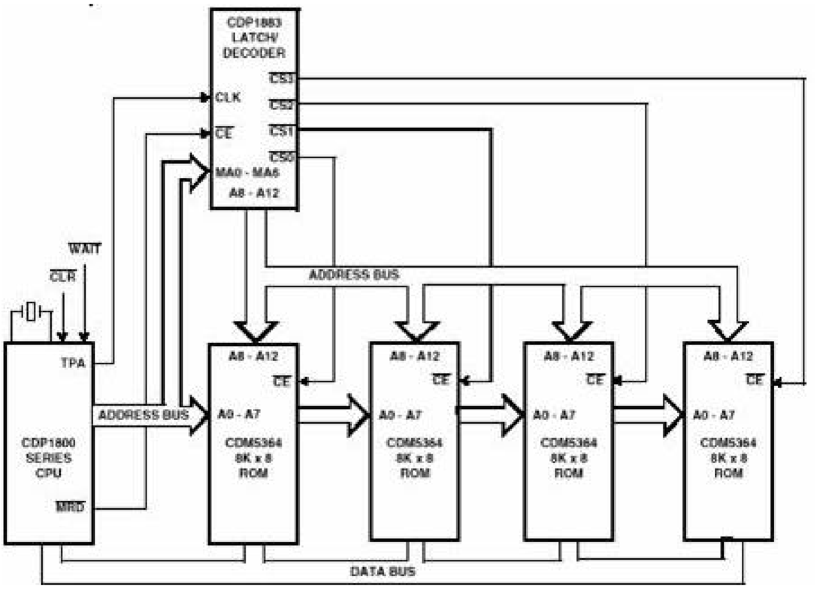


Figure 7: Typical CDP1802 ROM Layout

A CDP1883 latch decoder is used to select between an array of ROM chips. In this case the ROM chips are CDM5364 chips. The CDM5364 chips are 8K by 8 bits in size, and have roughly the same access time as the RAM. The CDP1883 latch is a 7-bit latch used to select between multitudes of different memory chips. The latch uses to the upper 7 bits of the address. The upper 2 high-order address bits are decoded by the latch and used to select between the ROM chips. The lower 5 high-order address bits coupled with all the low order bits are used to access a particular memory address within the selected ROM chip. [3]

PROM

An optional CMOS PROM can be added to the memory system of the 1802 architecture. The HM6641 chip is usually used for this type of memory. The HM6641 is a 512 byte by 8 bit CMOS PROM with a maximum access time of 250ns. This PROM has an integrated address latch, which allows easy interfacing to the multiplexed address bus of the 1802. [4]

Conclusion

While the RCA 1802 is considered by today's standards a simplistic example of a CMOS processor, it was considered cutting edge technology in 1974. With its single address space for both data and instructions and its single bi-directional data bus, the RCA 1802 can be classified as a Von-Neumann or Princeton architecture machine. The simplicity of the design and its low cost allowed for great flexibility and a cornucopia of applications. It has a large register set, 16 16-bit general purpose registers, and a small instruction set, 91 different instructions. These features along with its short instruction time were a milestone on the road to RISC chip design.

Bibliography

1. *CDP1802A,CDP1802AC,CDP1802BC*, no. 1, vol. 1, Intersil, http://www.intersil.com/data/fn/fn1305.pdf , Mar. 1997.

2. *CDP1824, CDP1824C*, no. 1, vol. 1, Intersil, http://www.intersil.com/data/FN/fn1103.pdf, Mar. 1997.

3. *CDP1883, CDP1883C*. no. 1, vol. 1, Intersil, http://www.intersil.com/data/FN/FN1/fn1507/FN1507.pdf, Mar. 1997.

4. Ramja, PR. “CMOS PROM Interfaces with COSMAC,” *Electronic Engineering*, vol. 57, no. 1, July 1985, pp. 36-38.

5. Unknown, “CDP1802 programming methods and mnemonic definitions.” *Electronic Design*, vol. 51, no. 1,May 24,1978, pp. 122-130.

6. Pittman, Tom. *A Short Course in Programming*, http://www.sbuniv.edu/~tpittman/IttyBitty/ShortCor.htm (Current 1980).

7. *CDP1851, CDP1851C*, no. 1, vol. 1, Intersil, http://www.intersil.com/Data/fn/fn1/fn1056/FN1056.pdf, Mar. 1997.

8. *RCA CMOS LSI Products CDP1861C*, no. 1, vol. 1, RCA, http://www.cosmacelf.com/cdp1861.pdf .

9. *CDP1871A, CDP1871AC*, no. 1, vol. 1, Harris Semiconductor, http://www.cosmacelf.com/cdp1871.pdf , Aug. 1996.

10. *IP-1802 Microprocessor Core*, no. 1, vol. 1, Sierra Circuit Design, Inc. home.teleport.com/~scd/IP/IP1802.pdf .

11. Joseph Weisbecker, “Build the COSMAC ‘ELF’: A Low-Cost Experimenter’s Microcomputer, Parts 1-4,” *Popular Electronics*, Aug. 1976 pgs 33-38, Sept. 1976 pgs 37-40, Mar. 1977 pgs 63-67, July 1977 pgs 41-46.

12. Online Computing Dictionary, http://www.instantweb.com/foldoc/foldoc.cgi?RCA+1802 (current Nov. 2002)

13. “COSMAC ELF: The 1802’s Place in Microcomputer History,” http://homepage.mac.com/ruske/cosmacelf/history1.htm (current Nov. 2002)

14. “The Antique Chip Collector’s Page,” http://www.antiquetech.com/chips/RCA1802.htm (current Nov. 2002)

15. http://groups.yahoo.com/group/cosmacelf/message/138 (current Nov. 2002)

16. *User Manual for the RCA CDP1802 COSMAC Microprocessor*, No.1, Vol. 1, RCA, 1977.