INTERSIL

IM6100 CMOS 12 Bit Microprocessor

FEATURES

- Silicon Gate Complementary MOS
- Fully Static - 0 to 5.7 MHz
- Single Power Supply
  IM6100 V_{CC} = 5 volts
  IM6100A V_{CC} = 10 volts
- Crystal Controlled On Chip Timing
- PDP®-8/e, Instruction Set Compatible
- Low Power Dissipation
  < 10mW @ 3.3 MHz @ 5 volts
- TTL Compatible at 5 volts
- Excellent Noise Immunity
- Direct Memory Access (DMA)
- Interrupt

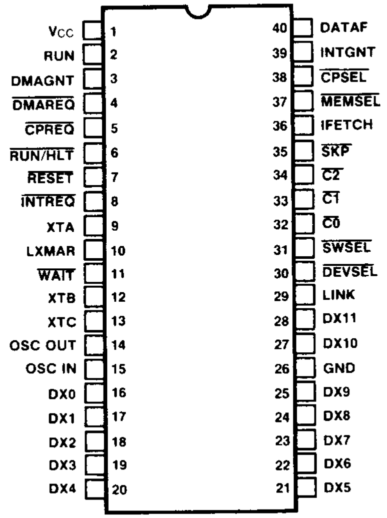
® PDP-8 is a registered trademark of Digital Electronics Corp.

GENERAL DESCRIPTION

The IM6100 is a fixed word length, single word instruction, parallel transfer microprocessor using 12-bit, two's complement arithmetic which recognizes the instruction set of Digital Equipment Corporation's PDP-8/e minicomputer. The internal circuitry is completely static and designed to operate at any speed between DC and the maximum operating frequency. Two pins are available to allow for an external crystal, thereby eliminating the need for clock generators and level translators. The crystal can be removed and the processor clocked by an external clock generator. The device design is optimized to minimize the number of external components required for interfacing with standard memory and peripheral devices.

The IM6100 family includes IM6101 (Programmable Interfacing Element), IM6102 (Memory Extension/DMA Controller/Interval Timer), IM6103 (Parallel Input-Output Port), IM6512 (64 x 12 RAM), IM6312 (1k x 12 ROM), and IM6402/03 (UART), all featuring ultra low power-high noise immunity CMOS characteristics. The entire family is supported by the 6910 Intercept II Microcomputer Development System.

PIN CONFIGURATION (outline dwg DL, PL)



BLOCK DIAGRAM

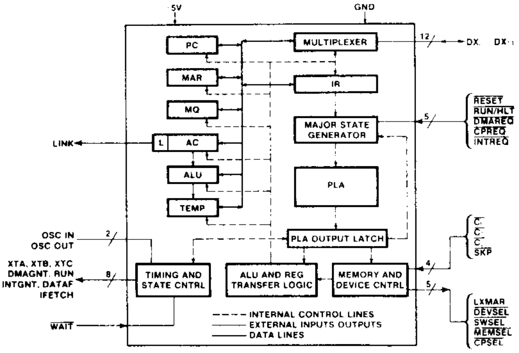


Figure 1: Functional Block Diagram

ORDERING INFORMATION

ORDER CODE	IM6100-1	IM6100A	IM6100
PLASTIC PKG.	IM6100-1IPL	IM6100-AIPL	IM6100-IPL
CERAMIC PKG.	IM6100-1IDL	IM6100-AIDL	IM6100-IDL
MILITARY TEMP.	IM6100-1MDL	IM6100-AMDL	IM6100-IDL
MILITARY TEMP. WITH 883B	IM6100-1MDL/883B	IM6100-AMDL/883B	—

IM6100

ABSOLUTE MAXIMUM RATINGS

Operating Temperature	
Industrial IM6100	−45°C to +85°C
Storage Temperature	−65°C to +150°C
Operating Voltage	+4.0V to +11.0V
Supply Voltage	+12.0V
Voltage On Any Input or Output Pin	−0.3V to V_{CC} +0.3V

NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent device failure. These are stress ratings only and functional operation of the devices at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may cause device failures.

D.C. CHARACTERISTICS

TEST CONDITIONS: V_{CC} = 5.0V ±10%, T_A = −40°C to +85°C

	SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
1	V_{IH}	Input Voltage High		V_{CC}−2.0			V
2	V_{IL}	Input Voltage Low				0.8	V
3	I_{IL}	Input Leakage	GND ≤ V_{IN} ≤ V_{CC}	−1.0		1.0	μA
4	V_{OH}	Output Voltage High	I_{OH} = −0.2mA	2.4			V
5	V_{OL}	Output Voltage Low	I_{OL} = 2.0mA			0.45	V
6	I_{OLK}	Output Leakage	GND ≤ V_{OUT} ≤ V_{CC}	−1.0		1.0	μA
7	I_{CC}	Power Supply Current-Standby	V_{IN} = GND or V_{CC}			800	μA
8	I_{CC}	Power Supply Current-Dynamic	f_C = 2.5MHz			1.8	mA
9	C_{IN}	Input Capacitance			7.0	8.0	pF
10	C_O	Output Capacitance			8.0	10.0	pF

A.C. CHARACTERISTICS (See Figure 2 and 22)

TEST CONDITIONS: V_{CC} = 5.0V ±10%, C_L = 50pF, T_A −40°C to +85°C, f_C = 2.5MHz

	SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
1	f_{op}	Operating Frequency			2.5	MHz
2	t_s	Major State Time	800			ns
3	t_{lxmar}	LXMAR Pulse Width	335			ns
4	t_{as}	Address Setup Time : DX-LXMAR (↓)	120			ns
5	t_{ah}	Address Hold Time : LXMAR (↓)-DX	175			ns
8	t_{end}	Data Output Enable Time: DEVSEL (↓)-DX			575	ns
6	t_{al}	Access Time from LXMAR			650	ns
7	t_{en}	Output Enable Time (MEM, CP, DEVSEL)			400	ns
9	t_{wp}	Pulse Width (MEMSEL, CPSEL)	320			ns
10	t_{wpd}	Pulse Width (DEVSEL)	320			ns
11	t_{ds}	Data Setup Time (DX- ↑ MEMSEL/CPSEL)	240			ns
12	t_{dh}	Data Hold Time (↑ MEMSEL/CPSEL-DX)	175			ns
13	t_{dsd}	Data Setup Time (DX-↑ DEVSEL)	275			ns
14	t_{dhd}	Data Hold Time (↑ DEVSEL-DX)	175			ns
15	t_{sl}	Logic Delay to MEM/DEV/CP/SWSEL	75		440	ns
16	t_{xt}	Logic Delay to LXMAR, XTA, XTB, XTC	65		380	ns
17	t_{st}	Logic Delay to DATAF, RUN, DMAGNT, INTGNT, LINK, IFETCH			475	ns
18	t_{rs}	Set up Time for CP/INT/DMAREQ	0			ns
19	t_{rh}	Hold Time for CP/INT/DMAREQ, RESET, RUN-HALT	300			ns
20	t_{rhp}	RUN-HALT Pulse Width	110			ns
21	t_{ws}	Set up Time for Wait	100			ns
22	t_{wh}	Hold Time for Wait	35			ns

Note: For capacitance greater than 50pF, the AC parameters will have a delay factor of 0.5ns/pF.

IM6100A

ABSOLUTE MAXIMUM RATINGS

Operating Temperature
Industrial IM6100AI .... -40°C to +85°C
Storage Temperature .... -65°C to +150°C
Operating Voltage .... +4.0V to +11.0V
Supply Voltage .... +12.0V
Voltage On Any Input or
Output Pin .... -0.3V to V_{CC} +0.3V

NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent device failure. These are stress ratings only and functional operation of the devices at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may cause device failures.

D.C. CHARACTERISTICS

TEST CONDITIONS: V_{CC} = 10V ±5%, T_A = -40°C to +85°C

	SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
1	V_{IH}	Input Voltage High		70% V_{CC}			V
2	V_{IL}	Input Voltage Low				20% V_{CC}	V
3	I_{IL}	Input Leakage	GND ≤ V_{IN} ≤ V_{CC}	-1.0		1.0	μA
4	V_{OH}	Output Voltage High	I_{OH} = 0.0mA	V_{CC} -0.01			V
5	V_{OL}	Output Voltage Low	I_{OL} = 0.0mA			GND +0.01	V
6	I_{OLK}	Output Leakage	GND ≤ V_{OUT} ≤ V_{CC}	-1.0		1.0	μA
7	I_{CC}	Power Supply Current-Standby	V_{IN} = GND or V_{CC}			900	μA
8	I_{CC}	Power Supply Current-Dynamic	f_C = 5.71MHz			4.0	mA
9	C_{IN}	Input Capacitance			7.0	8.0	pF
10	C_O	Output Capacitance			8.0	10.0	pF

A.C. CHARACTERISTICS (Ref: Figures 2 and 22)

TEST CONDITIONS: V_{CC} = 10V ±5%, C_L = 50pF, T_A = -40°C to +85°C, f_C = 5.71MHz

	SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
1	f_{op}	Operating Frequency			5.71	MHz
2	t_s	Major State Time	350			ns
3	t_{lxmar}	LXMAR Pulse Width	150			ns
4	t_{as}	Address Setup Time : DX-LXMAR (↓)	55			ns
5	t_{ah}	Address Hold Time : LXMAR (↓)-DX	60			ns
8	t_{end}	Data Output Enable Time: DEVSEL (↓)-DX			250	ns
6	t_{al}	Access Time from LXMAR			295	ns
7	t_{en}	Output Enable Time (MEM, CP, DEVSEL)			185	ns
9	t_{wp}	Pulse Width (MEMSEL, CPSEL)	140			ns
10	t_{wpd}	Pulse Width (DEVSEL)	140			ns
11	t_{ds}	Data Setup Time (DX- ↑ MEMSEL/CPSEL)	115			ns
12	t_{dh}	Data Hold Time (↑ MEMSEL/CPSEL-DX)	60			ns
13	t_{dsd}	Data Setup Time (DX- ↑ DEVSEL)	110			ns
14	t_{dhd}	Data Hold Time (↑ DEVSEL-DX)	60			ns
15	t_{sl}	Logic Delay to MEM/DEV/CP/SWSEL	35		180	ns
16	t_{xt}	Logic Delay to LXMAR, XTA, XTB, XTC	35		155	ns
17	t_{st}	Logic Delay to DATAF, RUN, DMAGNT, INTGNT, LINK, IFETCH			190	ns
18	t_{rs}	Set up Time for CP/INT/DMAREQ	0			ns
19	t_{rh}	Hold Time for CP/INT/DMAREQ, RESET, RUN-HALT	125			ns
20	t_{rhp}	RUN-HALT Pulse Width	45			ns
21	t_{ws}	Set up Time for Wait	45			ns
22	t_{wh}	Hold Time for Wait	15			ns

Note: For capacitance greater than 50pF, the AC parameters will have a delay factor of 0.5ns/pF.

IM6100AM (Military)

ABSOLUTE MAXIMUM RATINGS

Operating Temperature	
Industrial IM6100AM	−55°C to +125°C
Storage Temperature	−65°C to +150°C
Operating Voltage	+4.0V to +11.0V
Supply Voltage	+12.0V
Voltage On Any Input or Output Pin	−0.3V to V_{CC} +0.3V

NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent device failure. These are stress ratings only and functional operation of the devices at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may cause device failures.

D.C. CHARACTERISTICS

TEST CONDITIONS: V_{CC} = 10V ±5%, T_A = −55°C to +125°C

	SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
1	V_{IH}	Input Voltage High		70% V_{CC}			V
2	V_{IL}	Input Voltage Low				20% V_{CC}	V
3	I_{IL}	Input Leakage	GND ≤ V_{IN} ≤ V_{CC}	−1.0		1.0	μA
4	V_{OH}	Output Voltage High	I_{OH} = 0.0mA	V_{CC} −0.01			V
5	V_{OL}	Output Voltage Low	I_{OL} = 0.0mA			GND +0.01	V
6	I_{OLK}	Output Leakage	GND ≤ V_{OUT} ≤ V_{CC}	−1.0		1.0	μA
7	I_{CC}	Power Supply Current-Standby	V_{IN} = GND or V_{CC}			900	μA
8	I_{CC}	Power Supply Current-Dynamic	f_C = 5.0MHz			4.0	mA
9	C_{IN}	Input Capacitance			7.0	8.0	pF
10	C_O	Output Capacitance			8.0	10.0	pF

A.C. CHARACTERISTICS (Ref.: Figures 2 and 22)

TEST CONDITIONS: V_{CC} = 10V ± 5%, C_L = 50pF, T_A = −55°C to +125°C, f_C = 5.0MHz

	SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
1	f_{op}	Operating Frequency			5.0	MHz
2	t_s	Major State Time	400			ns
3	t_{lxmar}	LXMAR Pulse Width	170			ns
4	t_{as}	Address Setup Time : DX-LMAR (↓)	70			ns
5	t_{ah}	Address Hold Time : LXMAR (↓)-DX	70			ns
8	t_{end}	Data Output Enable Time: DEVSEL (↓)-DX			290	ns
6	t_{ai}	Access Time from LXMAR			340	ns
7	t_{en}	Output Enable Time (MEM, CP, DEVSEL)			220	ns
9	t_{wp}	Pulse Width (MEMSEL, CPSEL)	160			ns
10	t_{wpd}	Pulse Width (DEVSEL)	160			ns
11	t_{ds}	Data Setup Time (DX- ↑ MEMSEL/CPSEL)	140			ns
12	t_{dh}	Data Hold Time (↑ MEMSEL/CPSEL-DX)	70			ns
13	t_{dsd}	Data Setup Time (DX- ↑ DEVSEL)	140			ns
14	t_{dhd}	Data Hold Time (↑ DEVSEL-DX)	70			ns
15	t_{sl}	Logic Delay to MEM/DEV/CP/SWSEL	35		210	ns
16	t_{xt}	Logic Delay to LXMAR, XTA, XTB, XTC	35		170	ns
17	t_{st}	Logic Delay to DATAF, RUN, DMAGNT, INTGNT, LINK, IFETCH			210	ns
18	t_{rs}	Set up Time for CP/INT/DMAREQ	0			ns
19	t_{rh}	Hold Time for CP/INT/DMAREQ, RESET, RUN-HALT	140			ns
20	t_{rhp}	RUN-HALT Pulse Width	50			ns
21	t_{ws}	Set up Time for Wait	50			ns
22	t_{wh}	Hold Time for Wait	20			ns

Note: For capacitance o' greater than 50pF, the AC parameters will have a delay factor of 0.5ns/pF.

IM6100-1

ABSOLUTE MAXIMUM RATINGS

Operating Temperature
Industrial IM6100-1I .... −40°C to +85°C
Storage Temperature .... −65°C to +150°C
Operating Voltage .... +4.0V to +11.0V
Supply Voltage .... +12.0V
Voltage On Any Input or
Output Pin .... −0.3V to V_{CC} +0.3V

NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent device failure. These are stress ratings only and functional operation of the devices at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may cause device failures.

D.C. CHARACTERISTICS

TEST CONDITIONS: V_{CC} = 5.0V ±10%, T_A = −40°C to +85°C

	SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
1	V_{IH}	Input Voltage High		V_{CC} −2.0			V
2	V_{IL}	Input Voltage Low				0.8	V
3	I_{IL}	Input Leakage	GND ≤ V_{IN} ≤ V_{CC}	−1.0		1.0	μA
4	V_{OH}	Output Voltage High	I_{OH} = −0.2mA	2.4			V
5	V_{OL}	Output Voltage Low	I_{OL} = 2.0mA			0.45	V
6	I_{OLK}	Output Leakage	GND ≤ V_{OUT} ≤ V_{CC}	−1.0		1.0	μA
7	I_{CC}	Power Supply Current-Standby	V_{IN} = GND or V_{CC}			800	μA
8	I_{CC}	Power Supply Current-Dynamic	f_C = 3.33MHz			2.0	mA
9	C_{IN}	Input Capacitance			7.0	8.0	pF
10	C_O	Output Capacitance			8.0	10.0	pF

A.C. CHARACTERISTICS (Ref. Fig. 2 and 22)

TEST CONDITIONS: V_{CC} = 5.0V ± 10%, C_L = 50pF, T_A = −40°C to +85°C, f_C = 3.33MHz

	SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
1	t_{op}	Operating Frequency			3.33	MHz
2	t_s	Major State Time	600			ns
3	t_{lxmar}	LXMAR Pulse Width	260			ns
4	t_{as}	Address Setup Time: DX-LXMAR (↓)	85			ns
5	t_{ah}	Address Hold Time : LXMAR (↓)-DX	125			ns
8	t_{end}	Data Output Enable Time: DEVSEL (↓)-DX			470	ns
6	t_{al}	Access Time from LXMAR			520	ns
7	t_{en}	Output Enable Time (MEM, CP, DEVSEL)			300	ns
9	t_{wp}	Pulse Width (MEMSEL, CPSEL)	235			ns
10	t_{wpd}	Pulse Width (DEVSEL)	235			ns
11	t_{ds}	Data Setup Time (DX-↑ MEMSEL/CPSEL)	135			ns
12	t_{dh}	Data Hold Time (↑ MEMSEL/CPSEL-DX)	125			ns
13	t_{dsd}	Data Setup Time (DX-↑ DEVSEL)	225			ns
14	t_{dhd}	Data Hold Time (↑ DEVSEL-DX)	125			ns
15	t_{sl}	Logic Delay to MEM/DEV/CP/SWSEL	75		380	ns
16	t_{xt}	Logic Delay to LXMAR, XTA, XTB, XTC	65		270	ns
17	t_{st}	Logic Delay to DATAF, RUN, DMAGNT, INTGNT, LINK, IFETCH			340	ns
18	t_{rs}	Set up Time for CP/INT/DMAREQ	0			ns
19	t_{rh}	Hold Time for CP/INT/DMAREQ, RESET, RUN-HALT	200			ns
20	t_{rhp}	RUN-HALT Pulse Width	80			ns
21	t_{ws}	Set up Time for Wait	100			ns
22	t_{wh}	Hold Time for Wait	20			ns

Note: For capacitance greater than 50pF, the AC parameters will have a delay factor of 0.5ns/pF.

IM6100-1M (Military)

ABSOLUTE MAXIMUM RATINGS

Operating Temperature	
Industrial IM6100-1M	−55° C to +125° C
Storage Temperature	−65° C to +150° C
Operating Voltage	+4.0V to +11.0V
Supply Voltage	+12.0V
Voltage On Any Input or Output Pin	−0.3V to V_{CC} +0.3V

NOTE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent device failure. These are stress ratings only and functional operation of the devices at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may cause device failures.

D.C. CHARACTERISTICS

TEST CONDITIONS: V_{CC} = 5V ± 10%, T_A = −55° C to +125° C

	SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
1	V_{IH}	Input Voltage High		V_{CC} −2.0			V
2	V_{IL}	Input Voltage Low				0.8	V
3	I_{IL}	Input Leakage	GND ≤ V_{IN} ≤ V_{CC}	−1.0		1.0	µA
4	V_{OH}	Output Voltage High	I_{OH} = −0.2mA	2.4			V
5	V_{OL}	Output Voltage Low	I_{OL} = 2.0mA			0.45	V
6	I_{OLK}	Output Leakage	GND ≤ V_{OUT} ≤ V_{CC}	−1.0		1.0	µA
7	I_{CC}	Power Supply Current-Standby	V_{IN} = GND or V_{CC}			800	µA
8	I_{CC}	Power Supply Current-Dynamic	f_C = 2.5MHz			2.0	mA
9	C_{IN}	Input Capacitance			7.0	8.0	pF
10	C_O	Output Capacitance			8.0	10.0	pF

A.C. CHARACTERISTICS (Ref. Fig. 2 and 22)

TEST CONDITIONS: V_{CC} = 5.0V ± 10%, C_L = 50pF, T_A = −55° C to +125° C, f_C = 2.5MHz

	SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
1	f_{op}	Operating Frequency			2.5	MHz
2	t_s	Major State Time	800			ns
3	t_{lxmar}	LXMAR Pulse Width	355			ns
4	t_{as}	Address Setup Time : DX-LXMAR (↓)	200			ns
5	t_{ah}	Address Hold Time : LXMAR (↓)-DX	175			ns
8	t_{end}	Data Output Enable Time: DEVSEL (↓)-DX			655	ns
6	t_{al}	Access Time from LXMAR			745	ns
7	t_{en}	Output Enable Time (MEM, CP, DEVSEL)			470	ns
9	t_{wp}	Pulse Width (MEMSEL, CPSEL)	330			ns
10	t_{wpd}	Pulse Width (DEVSEL)	330			ns
11	t_{ds}	Data Setup Time (DX- ↑ MEMSEL/CPSEL)	250			ns
12	t_{dh}	Data Hold Time (↑ MEMSEL/CPSEL-DX)	170			ns
13	t_{dsd}	Data Setup Time (DX- ↑ DEVSEL)	350			ns
14	t_{dhd}	Data Hold Time (↑ DEVSEL-DX)	170			ns
15	t_{sl}	Logic Delay to MEM/DEV/CP/SWSEL	75		420	ns
16	t_{xt}	Logic Delay to LXMAR, XTA, XTB, XTC	65		300	ns
17	t_{st}	Logic Delay to DATAF, RUN, DMAGNT, INTGNT, LINK, IFETCH			375	ns
18	t_{rs}	Set up Time for CP/INT/DMAREQ	0			ns
19	t_{rh}	Hold Time for CP/INT/DMAREQ, RESET, RUN-HALT	220			ns
20	t_{rhp}	RUN-HALT Pulse Width	90			ns
21	t_{ws}	Set up Time for Wait	110			ns
22	t_{wh}	Hold Time for Wait	20			ns

Note: For capacitance of greater than 50pF, the AC parameters all have delay factor of 0.5ns/pF.

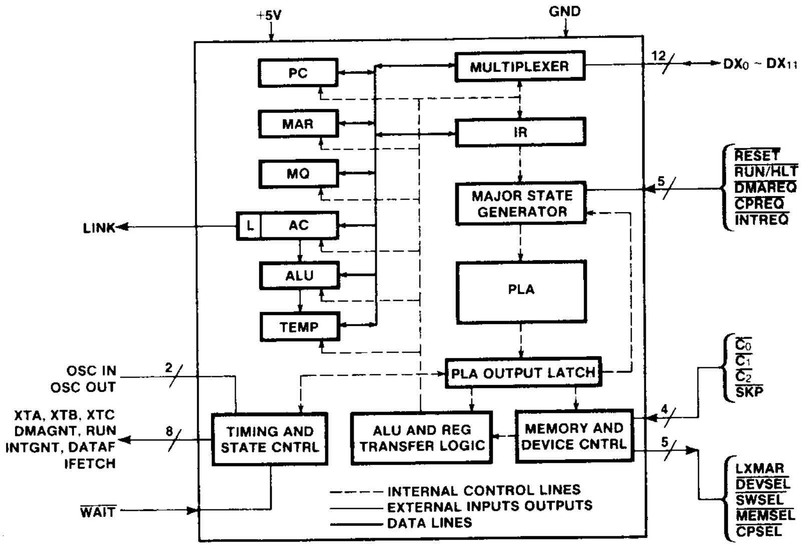


Figure 1: Functional Block Diagram

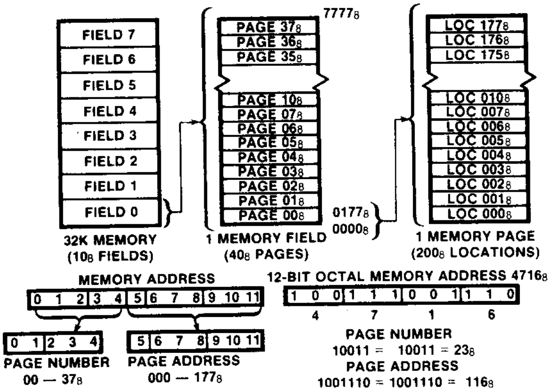
FUNCTIONAL PIN DESCRIPTIONS

PIN	SYMBOL	DESCRIPTION
1	V_{CC}	Supply voltage.
2	RUN	The signal indicates the runstate of the CPU and may be used to power down the external circuitry
3	DMAGNT	Direct Memory Access Grant—DX lines are three-state.
4	$\overline{\text{DMAREQ}}$	Direct Memory Access Request—DMA is granted at the end of the current instruction. Upon DMA grant, the CPU suspends program execution until the DMAREQ line is released.
5	$\overline{\text{CPREQ}}$	Control Panel Request—a dedicated interrupt which bypasses the normal device interrupt request structure.
6	$\overline{\text{RUN/HLT}}$	Pulsing the Run/Halt line causes the CPU to alternately run and halt by changing the state of the internal RUN/HLT flip flop.
7	$\overline{\text{RESET}}$	Clears the AC and loads 7777_8 into the PC. CPU is halted.
8	$\overline{\text{INTREQ}}$	Peripheral device interrupt request.
9	XTA	External coded minor cycle timing—signifies input transfers to the IM6100.
10	LXMAR	The Load External Memory Address Register is used to store memory and peripheral addresses externally.
11	$\overline{\text{WAIT}}$	Indicates that peripherals or external memory is not ready to transfer data. The CPU state gets extended as long as WAIT is active. The CPU is in the lowest power state with clocks running.
12	XTB	External coded minor cycle timing—signifies output transfers from the IM6100.
13	XTC	External coded minor cycle timing—used in conjunction with the Select Lines to specify read or write operations.
14	OSC OUT	Crystal input to generate the internal timing (also external clock input).
15	OSC IN	See Pin 14—OSC OUT (also external clock ground)
16	DX_0	DataX—multiplexed data in, data out and address lines.
17	DX_1	See Pin 16—DX_0.
18	DX_2	See Pin 16—DX_0.
19	DX_3	See Pin 16—DX_0.
20	DX_4	See Pin 16—DX_0.
21	DX_5	See Pin 16—DX_0.
22	DX_6	See Pin 16—DX_0.
23	DX_7	See Pin 16—DX_0.
24	DX_8	See Pin 16—DX_0.
25	DX_9	See Pin 16—DX_0.
26	GND	Ground
27	DX_{10}	See Pin 16—DX_0.
28	DX_{11}	See Pin 16—DX_0.
29	LINK	Indicates state of link flip flop.
30	$\overline{\text{DEVSEL}}$	Device Select for I/O transfers.
31	$\overline{\text{SWSEL}}$	Switch Register Select for the OR THE SWITCH REGISTER INSTRUCTION (OSR). OSR is a Group 2 Operate Instruction which reads a 12 bit external switch register and OR's it with the contents of the AC.
32	$\overline{C_0}$	Control line inputs from the peripheral device during an I/O transfer (Table VI).
33	$\overline{C_1}$	See Pin 32—$\overline{C_0}$.
34	$\overline{C_2}$	See Pin 32—$\overline{C_0}$.
35	SKP	Skips the next sequential instruction if active during an I/O instruction.
36	$\overline{\text{IFETCH}}$	Instruction Fetch Cycle
37	$\overline{\text{MEMSEL}}$	Memory Select for memory transfers.
38	$\overline{\text{CPSEL}}$	The Control Panel Memory Select becomes active, instead of the MEMSEL, for control panel routines. Signal may be used to distinguish between control panel and main memories.
39	INTGNT	Peripheral device Interrupt Grant.
40	DATAF	Data Field pin indicates the execute phase of indirectly addressed AND, TAD, ISZ and DCA instructions so that the data transfers are controlled by the Data Field, DF, and not the Instruction Field, IF, if Extended Memory Control hardware is used to extend the addressing space from 4K to 32K words.

MEMORY ORGANIZATION

The IM6100 has a basic addressing capacity of 4096 12-bit words which may be extended by Extended Memory Control hardware to 32K. The memory system is organized in 4096 word blocks, called MEMORY FIELDS. The first 4096 words of memory are in Field 0; if a full 32K of memory is installed, the uppermost Memory Field will be numbered 7. In any given Memory Field every location has a unique 4 digit octal (12 bit binary) address, 0000_8 to 7777_8 (0000_{10} to 4095_{10}). Each Memory Field is subdivided into 32 PAGES of 128 words each. Memory Pages are numbered sequentially from Page 00_8, containing addresses $0000\text{-}0177_8$, to Page 37_8, containing addresses $7600_8\text{-}7777_8$. The first 5 bits of a 12-bit MEMORY ADDRESS denote the PAGE NUMBER and the low order 7 bits specify the PAGE ADDRESS of the memory location within the given Page.

During an instruction fetch cycle, the IM6100 fetches the instruction pointed to by the PC, the contents of the PC are transferred to the MAR, and the PC is incremented by 1. The PC now contains the address of the 'next' sequential instruction and the MAR contains the address of the 'current' instruction which must be fetched from memory. Bits 0-4 of the MAR identify the CURRENT PAGE, that is, the Page from which instructions are currently being fetched, and bits 5-11 of the MAR identify the location within the Current Page. (PAGE ZERO (0), by definition, denotes the first 128 words of memory, $0000_8\text{-}0177_8$.)



Memory Organization

INSTRUCTION SET

The IM6100 instructions are 12-bit words stored in memory. The IM6100 makes no distinction between instructions and data; it can manipulate instructions as stored variables or execute data as instructions when it is programmed to do so. There are three general classes of IM6100 instructions. They are referred to as Memory Reference Instruction (MRI), Operate Instruction (OPR) and Input/Output Transfer Instruction (IOT).

The notations used in the following instruction tables are defined in Table I below:

TABLE 1. Notation Definitions

1. ( ) denotes the contents of the register or location within parenthesis. (EA) is read as "... the contents of the Effective Address."
2. (()) denotes the contents of the location pointed to by the contents of the location within the double parenthesis. ((PA)) is read as "... the contents of the location pointed to by the contents of the Pointer Address."
3. – denotes "... is replaced by ..."
4. – denotes the interchange operation.
5. ∧ denotes logical AND operation.
6. ∨ denotes logical OR operation.
7. EA denotes the Effective Address for Direct Addressing.
8. PA denotes the Pointer Address for Indirect Addressing. PA can be any address on the CURRENT PAGE or PA can be any address (0000_8) through (0177_8) on PAGE ZERO other than the addresses (0010_8) through (0017_8) which are reserved for autoindexing.
9. PAIX denotes the Pointer Address for autoindexing. It can be any address (0010_8) through (0017_8).
10. I represents bit 3, the Indirect Addressing Bit, of the instruction.
11. EA, PA, or PAIX is specified by bit 4 through bit 11 of the memory reference instruction.
12. PC denotes the Program Counter.
13. SR denotes the Switch Register.
14. (AC)n denotes the nth bit of the AC contents.
15. DEV denotes a specific peripheral device and "ddddd" denotes the device address code. CMND is the command issued to the device during an I/O operation and "eee" is its three bit code.

ARCHITECTURE

The IM6100 has 6 twelve bit registers, a programmable logic array, an arithmetic and logic unit and associated gating and timing circuitry. A block diagram of the IM6100 is shown in Figure 1.

ACCUMULATOR (AC)

The AC is a 12-bit register in which arithmetic and logical operations are performed. Data words may be transferred from memory to the AC or transferred from the AC into memory. Arithmetic and logical operations involve one or two operands, one held in the AC and the other fetched from the memory. The result of the operation is left in the AC which may be cleared, complemented, tested, incremented or rotated under program control. The AC also serves as an input-output register, as all programmed data transfers pass through the AC.

LINK (L)

The Link is a 1-bit flip-flop that serves as a high-order extension of the AC. It is used as a carry flip-flop for 2's complement arithmetic. A carry out of the accumulator complements the Link. Link can be cleared, set, complemented and tested under program control and rotated as part of the AC.

MQ REGISTER (MQ)

The MQ is a 12-bit temporary register which is program accessible. The contents of AC may be transferred to the MQ for temporary storage, or MQ can be OR'ed with the AC and the result stored in the AC. The contents of the AC and the MQ may also be exchanged.

MEMORY ADDRESS REGISTER (MAR)

While accessing memory, the 12-bit MAR register contains the address of the memory location that is currently selected for reading or writing. The MAR is also used as an internal register for microprogram control during data transfers to and from memory and peripherals.

PROGRAM COUNTER (PC)

The 12-bit PC contains the address of the memory location from which the next instruction is fetched. During an instruction fetch, the PC is transferred to MAR and the PC is then incremented by 1. When there is a branch to another address in memory, the branch address is set into the PC. Branching normally takes place under program control, however, during an input-output operation, a device may specify a branch address. A skip (SKP) instruction increments the PC by 1, thus causing the next instruction to be skipped. The SKP instruction may be unconditional, or conditional on the state of the AC or the Link. During an input-output operation, a device can also cause the next sequential instruction to be skipped. Interrupts force the PC to 0000. Reset forces the PC to 7777_8.

ARITHMETIC AND LOGICAL UNIT (ALU)

The ALU performs both arithmetic and logical operations, –two's complement binary addition, AND, OR and complement. The ALU can perform a single position shift either to the left or to the right; a double rotate is implemented in two single bit shifts. The ALU can also shift by 3 positions to implement a byte swap in two steps. The AC is always one of the inputs to the ALU, however, under internal microprogram control, AC may be gated off and all one's or all zero's gated in. The second input may be any one of the other registers under internal microprogram control.

TEMPORARY REGISTER (TEMP)

The 12-bit TEMP register latches the result of an ALU operation, before it is sent to the destination register, to avoid race conditions. The TEMP is also used as an internal register for microprogram control.

INSTRUCTION REGISTER (IR)

During an instruction fetch, the 12-bit IR is loaded with the instruction that is to be executed by the CPU. The IR specifies the initial step of the microprogram sequence for each instruction, and is also used as an internal register to store temporary data for microprogram control.

MULTIPLEXER (DX)

The 12-bit Input/Output Multiplexer handles data, address and instruction transfers into and out of the CPU, and to or from the main memory and peripheral devices on a time-multiplexed basis.

MAJOR STATE GENERATOR AND THE PROGRAMMED LOGIC ARRAY (PLA)

During an instruction fetch the instruction to be executed is loaded into the IR. The PLA is then used for the correct sequencing of the CPU for the appropriate instruction. After an instruction is completely sequenced, the major state generator scans the internal priority network, which decides whether the machine is going to fetch the next instruction in sequence, or service one of the external request lines.

PLA OUTPUT LATCH

The PLA Output Latch permits the PLA to be pipelined; it fetches the next control sequence while the CPU is executing the current sequence.

MEMORY AND DEVICE CONTROL, ALU AND REG TRANSFER LOGIC

The Memory and Device Control Unit provides external control signals to communicate with peripheral devices (DEVSEL), switch register (SWSEL), memory (MEMSEL) and/or control panel memory (CPSEL). During I/O instructions this unit also modifies the PLA outputs depending on the states of the four device control lines (SKP, C_0, C_1, C_2). The ALU and Register Transfer Logic provides the control signals for the internal register transfers and ALU operation.

ARCHITECTURE (CONTINUED)

TIMING AND STATE CONTROL

The IM6100 internally generates all the timing and state signals. A crystal is used to control the CPU operating frequency, which is divided by two by the CPU. With a 4MHz crystal, the internal states will be of 500nsec duration. The major timing states are described in Figure 2.

T_1 For memory reference instructions, a 12-bit address is sent on the DX lines. The Load External Memory Address Register, LXMAR, is used to clock an external register to store the address information externally, if required. When executing an Input-Output I/O instruction, the instruction being executed is sent on the DX lines to be stored externally. The external address register then contains the device address and control information. The LXMAR pulse occurs only if a valid address is present on the DX lines.

Various CPU request lines are priority sampled if the next cycle is an Instruction Fetch cycle. Current state of the CPU is available externally.

T_2 Memory/Peripheral data is read for an input transfer (READ). WAIT controls the transfer duration. If WAIT is active during input transfers, the CPU waits in the T_2 state. The wait duration is an integral multiple of the crystal frequency — 250nsec for 4MHz.

For memory reference instructions, the Memory Select, MEMSEL, line is active. For I/O instructions the Device Select, DEVSEL, line is active. Control lines, therefore, distinguish the contents of the external register as memory or device address.

External device sense lines, C_0, C_1, C_2, and SKP, are sampled if the instruction being executed is an I/O instruction.

Control Panel Memory Select, CPSEL, and Switch Register Select, SWSEL, become active low for data transfers between the IM6100 and Control Panel Memory and the Switch Register, respectively.

T_3,T_4,T_5 ALU operation and internal register transfers.

T_6 This state is entered for an output transfer (WRITE). The address is defined during T_1. WAIT controls the time for which the Write data must be maintained.

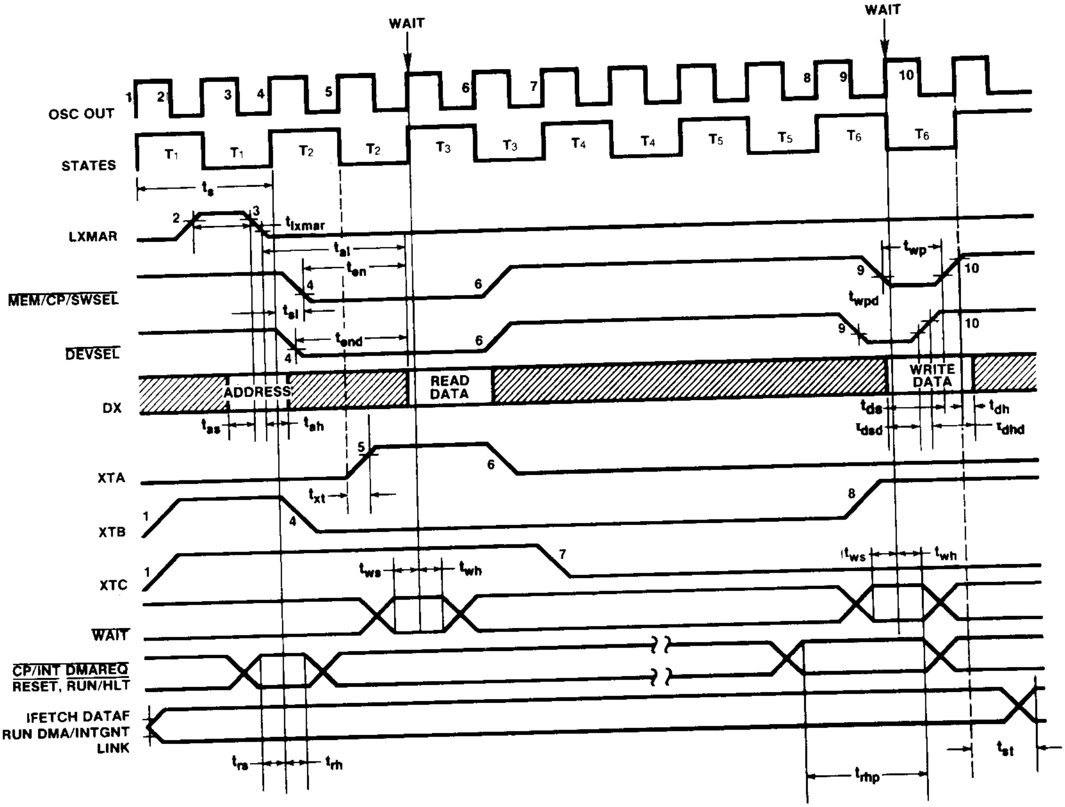


Figure 2: IM6100 AC Timing Diagram

INSTRUCTION SET

MEMORY REFERENCE INSTRUCTION (MRI)

The Memory Reference Instructions operate on the contents of a memory location or use the contents of a memory location to operate on the AC or the PC. The first 3 bits of a Memory Reference Instruction specify the operation code, or OPCODE, and the low order 9 bits, the OPERAND address, as shown in Figure 3.

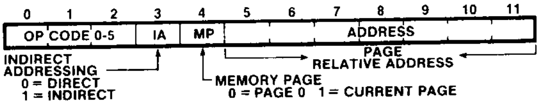


Figure 3: Memory Reference Instruction Format

Bits 5 through 11, the PAGE ADDRESS, identify the location of the OPERAND on a given page, but they do not identify the page itself. The page is specified by bit 4, called the CURRENT PAGE OR PAGE 0 BIT. If bit 4 is a 0, the page address is interpreted as a location on Page 0. If bit 4 is a 1, the page address specified is interpreted to be on the Current Page.

For example, if bits 5 through 11 represent 123_8 and bit 4 is a 0, the location referenced is the absolute address 0123_8. However, if bit 4 is a 1 and the current instruction is in a memory location whose absolute address is 4610_8 the page address 123_8 designates the absolute address 4723_8, as shown below.

4610_8 = 100 110 001 000 = PAGE 10 011 = PAGE 23_8
Location 4610_8 is in PAGE 23_8. Location 123_8 in PAGE 23_8, CURRENT PAGE, will be:

$$\underbrace{\underbrace{10\ 011}_{}\ \underbrace{1\ 010\ 011}_{}} = 100\ 111\ 010\ 011 = 4723_8$$

PAGE ADDRESS 123_8
PAGE NUMBER 23_8

By this method, 256 locations may be directly addressed, 128 on PAGE 0 and 128 on the CURRENT PAGE. Other locations are addressed indirectly by setting bit 3. An INDIRECT ADDRESS (pointer address) identifies the location that contains the desired address (effective address). To address a location that is not directly addressable, not in PAGE 0 or in the CURRENT PAGE, the absolute address of the desired location is stored in one of the 256 directly addressable locations (pointer address). Upon execution, the MRI will operate on the contents of the location identified by the address contained in the pointer location.

It should be noted that locations 0010_8-0017_8 in PAGE 0 are AUTOINDEXED. If these locations are addressed indirectly, the contents are incremented by 1 and restored before they are used as the operand address. These locations may, therefore, be used for indexing applications.

Table II lists the mnemonics for the six memory reference instructions, their OPCODEs, the operations they perform and the number of states required for execution.

It should be noted that the data is represented in Two's Complement Integer notation. In this system, the negative of a number is formed by complementing each bit in the data word and adding "1" to the complemented number. The sign is indicated by the most significant bit. In the 12-bit word used by the IM6100, when bit 0 is a "0", it denotes a positive number and when bit 0 is a "1", it denotes a negative number. The maximum single precision number ranges for this system are 3777_8 (+2047) and 4000_8 (−2048).

Table II

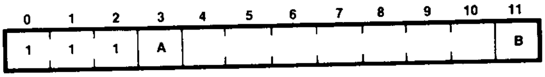
MNEMONIC	OP CODE	IA	STATES	OPERATION
AND EA	0_8	0	10	LOGICAL AND DIRECT Operation (AC) ← (AC) ∧ (EA) Description Contents of the EA are logically AND'ed with the contents of the AC and the result is stored in AC.
AND I PA		1	15	LOGICAL AND INDIRECT (PA ≠ 0010-0017_8) Operation (AC) ← (AC) ∧ ((PA))
AND PAIX		1	16	LOGICAL AND AUTOINDEX (PAIX = 0010-0017_8) Operation (PA) ← (PA) + 1, (AC) ← (AC) ∧ ((PA))
TAD EA	1_8	0	10	BINARY ADD DIRECT Operation (AC) ← (AC) + (EA) Description Contents of the EA are ADD'ed with the contents of the AC and the result is stored in the AC, carry out complements the LINK. If AC is initially cleared, this instruction acts as LOAD from Memory.
TAD I PA		1	15	BINARY ADD INDIRECT (PA ≠ 0010-0017_8) Operation AC) ← (AC) + ((PA))
TAD I PAIX		1	16	BINARY ADD AUTOINDEX (PAIX = 0010-0017_8) Operation (PA) ← PA) + 1, (AC) ← (AC) + ((PA))
ISZ EA	2_8	0	16	INCREMENT AND SKIP IF ZERO DIRECT Operation. EA) ← (EA) + 1, if (EA) = 0000_8, PC ← PC + 1 Description. Contents of the EA are incremented by 1 and restored. If the result is zero, the next sequential instruction is skipped
ISZ I PA		1	21	INCREMENT AND SKIP IF ZERO INDIRECT (PA ≠ 0010-0017_8) Operation. ((PA)) ← ((PA)) + 1, if ((PA)) = 0000_8, PC ← PC + 1
ISZ I PAIX		1	22	INCREMENT AND SKIP IF ZERO AUTOINDEX (PAIX = 0010-0017_8) Operation (PA) ← (PA) + 1, ((PA)) ← ((PA)) + 1, if ((PA)) = 0000_8, PC ← PC + 1
DCA EA	3_8	0	11	DEPOSIT AND CLEAR THE ACCUMULATOR DIRECT Operation (EA) ← (AC), (AC) ← 0000_8 Description: The contents of the AC are stored in EA and the AC is cleared.
DCA I PA		1	16	DEPOSIT AND CLEAR THE ACCUMULATOR INDIRECT (PA ≠ 0010-0017_8) Operation ((PA)) ← (AC), (AC) ← 0000_8
DCA I PAIX		1	17	DEPOSIT AND CLEAR THE ACCUMULATOR AUTOINDEX (PAIX = 0010-0017_8) Operation PA) ← PA + 1 ((PA)) ← (AC), AC ← 0000_8
JMS EA	4_8	0	11	JUMP TO SUBROUTINE DIRECT Operation EA) ← (PC), PC) ← EA + 1 Description The contents of the PC are stored in the EA The PC is incremented by 1 immediately after every instruction fetch The contents of the EA now point to the next sequential instruction following the JMS (return address) The next instruction is taken from EA + 1
JMS I PA		1	16	JUMP TO SUBROUTINE INDIRECT (PA ≠ 0010-0017_8) Operation ((PA)) ← PC, (PC) ← (PA) + 1
JMS I PAIX		1	17	JUMP TO SUBROUTINE AUTOINDEX (PAIX = 0010-0017_8) Operation (PA) ← (PA) + 1, ((PA)) ← PC, (PC) ← (PA) + 1
JMP EA	5_8	0	10	JUMP DIRECT Operation (PC ← EA Description: The next instruction is taken from the EA.
JMP I PA		1	15	JUMP INDIRECT (PA ≠ 0010-0017_8) Operation (PC) (PA)
JMP I PAIX		1	16	JUMP AUTOINDEX (PAIX = 0010-0017_8) Operation (PA) + 1, (PC) ← (PA)

INSTRUCTION SET (CONTINUED)

OPERATE INSTRUCTIONS

The Operate Instructions, which have an OPCODE of 7_8 (111), consist of 3 groups of microinstructions. Group 1, which is identified by the presence of the 0 in bit 3, is used to perform logical operations on the contents of the accumulator and link. Group 2, which is identified by the presence of a 1 in bit 3 and a 0 in bit 11, is used primarily to test the contents of the Accumulator and/or Link and then conditionally skip the next sequential instruction. Group 3 has a 1 in bit 3 and a 1 in bit 11 and performs logical operations on the contents of the AC and MQ.

The basic OPR instruction format is shown in Figure 4.



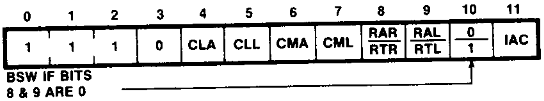
MICROINSTRUCTION	A	B
GROUP 1	0	0/1
GROUP 2	1	0
GROUP 3	1	1

Figure 4: Basic OPR Instruction Format

Operate microinstructions from any group may be microprogrammed with other operate microinstructions of the same group providing the instruction codes do not conflict. The actual code for a microprogrammed combination of two, or more, microinstructions is the bitwise logical OR of the octal codes for the individual microinstructions. When more than one operation is microprogrammed into a single instruction, the operations are performed in a prescribed sequence, with logical sequence number 1 performed first, logical sequence number 2 performed second, logical sequence number 3 performed third and so on. Two operations with the same logical sequence number, within a given group of microinstructions, are performed simultaneously.

GROUP MICROINSTRUCTIONS

Figure 5 shows the instruction format of a group 1 microinstruction. Any one of bits 4 to 11 may be set, loaded with a binary 1, to indicate a specific group 1 microinstruction. If more than one of these bits is set, the instruction is a microprogrammed combination of group 1 microinstructions, which will be executed according to the logical sequence shown in Figure 5.



LOGICAL SEQUENCES:
1—CLA, CLL
2—CMA, CML
3—IAC
4—RAR, RAL, RTR, RTL, BSW

Figure 5: Group 1 Microinstruction Format

Table III lists commonly used group 1 microinstructions, their assigned mnemonics, octal code, logical sequence, the number of states, and the operation they perform. The same format is followed in Table IV and V which lists group 2 and 3 microinstructions, respectively.

Table III: Group 1 Operate Microinstructions

MNEMONIC	OCTAL CODE	LOGICAL SEQUENCE	NUMBER OF STATES	OPERATION
NOP	7000	1	10	NO OPERATION—This instruction causes a 10 state delay in program execution, without affecting the state of the IM6100. It may be used for timing synchronization or as a convenient means of deleting an instruction from a program.
IAC	7001	3	10	INCREMENT ACCUMULATOR—The content of the AC is incremented by one (1) and carry out complements the Link (L).
RAL	7004	4	15	ROTATE ACCUMULATOR LEFT—The contents of the AC and L are rotated one binary position to the left. AC (0) is shifted to L and L is shifted to AC (11).
RTL	7006	4	15	ROTATE TWO LEFT—The contents of the AC and L are rotated two binary positions to the left. AC (1) is shifted to L and L is shifted to AC (10).
RAR	7010	4	15	ROTATE ACCUMULATOR RIGHT—The content of the AC and L are rotated one binary position to the right. AC (11) is shifted to L and L is shifted to AC (0).
RTR	7012	4	15	ROTATE TWO RIGHT—The contents of the AC and L are rotated two binary positions to the right. AC (10) is shifted to L and L is shifted to AC (1).
BSW	7002	4	15	BYTE SWAP—The right six (6) bits of the AC are exchanged or SWAPPED with the left six bits. AC (0) is swapped with AC (6), AC (1) with AC (7), etc. L is not affected.
CML	7020	2	10	COMPLEMENT LINK—The content of the link is complemented.
CMA	7040	2	10	COMPLEMENT ACCUMULATOR—The content of each bit of the AC is complemented having the effect of replacing the content of the AC with its one's complement.
CIA	7041	2,3	10	COMPLEMENT AND INCREMENT ACCUMULATOR—The content of the AC is replaced with its two's complement. Carry out complements the LINK.
CLL	7100	1	10	CLEAR LINK—The link is loaded with a binary 0.
CLL RAL	7104	1,4	15	CLEAR LINK—ROTATE ACCUMULATOR LEFT.
CLL RTL	7106	1,4	15	CLEAR LINK—ROTATE TWO LEFT.
CLL RAR	7110	1,4	15	CLEAR LINK—ROTATE ACCUMULATOR RIGHT.
CLL RTR	7112	1,4	15	CLEAR LINK—ROTATE TWO RIGHT.
STL	7120	1,2	10	SET THE LINK—The LINK is loaded with a binary 1 corresponding with a microprogrammed combination of CLL and CML.
CLA	7200	1	10	CLEAR ACCUMULATOR—The accumulator is loaded with binary 0's.
CLA IAC	7201	1,3	10	CLEAR ACCUMULATOR—INCREMENT ACCUMULATOR.
GLT	7204	1,4	15	GET THE LINK—The AC is cleared; the content of L is shifted into AC (11), a 0 is shifted into L. This is a microprogrammed combination of CLA and RAL.
CLA CLL	7300	1	10	CLEAR ACCUMULATOR—CLEAR LINK.
STA	7240	1,2	10	SET THE ACCUMULATOR—Each bit of the AC is set to 1 corresponding to a microprogrammed combination of CLA and CMA.

INSTRUCTION SET (CONTINUED)

GROUP 2 MICROINSTRUCTIONS

Figure 6 shows the instruction format of group 2 microinstructions. Bits 4-10 may be set to indicate a specific group 2 microinstruction. If more than one of bits 4-7 or 9-10 is set, the instruction is a microprogrammed combination of group 2 microinstructions, which will be executed according to the logical sequence shown in Figure 6.

Skip microinstructions may be microprogrammed with CLA, OSR, or HLT microinstructions. When two or more skip microinstructions are microprogrammed into a single instruction, the resulting condition on which the decision will be based is the logical OR of the individual conditions when bit 8 is 0, or, when bit 8 is 1, the decision will be based on the logical AND.

By combining skip instructions properly, all possible relational conditions can be tested (i.e., =, ≠, <, ≤, >, ≥). Skip microinstructions which have a 0 in bits 5, 6, 7, or 8 may not be microprogrammed with skip microinstructions which have a 1 in those same bits.

0	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	CLA	SMA SPA	SZA SNA	SNL SZL	0 1	OSR	HLT	0

LOGICAL SEQUENCES:
1 (BIT 8 IS ZERO)—SMA OR SZA OR SNL
(BIT 8 IS ONE) —SPA AND SNA AND SZL
2 —CLA
3 —OSR, HLT

Figure 6: Group 2 Microinstruction Format

Table IV: Group 2 Operate Microinstructions

MNEMONIC	OCTAL CODE	LOGICAL SEQUENCE	NUMBER OF STATES	OPERATION
NOP	7400	1	10	NO OPERATION—See Group 1 MICROINSTRUCTIONS
HLT	7402	3	10	HALT—Program stops at the conclusion of the current machine cycle If HLT is combined with others in OPR 2, the other operations are completed before the end of the cycle.
OSR	7404	3	15	OR WITH SWITCH REGISTER—The content of the Switch Register if OR'ed with the content of the AC and the result is stored in the AC. The OSR INSTRUCTION TIMING is shown in Figure 7. The IM6100 sequences the OSR instruction through a 2-cycle execute phase referred to as OPR 2A and OPR 2B.
SKP	7410	1	10	SKIP—The content of the PC is incremented by 1, to skip the next sequential instruction
SNL	7420	1	10	SKIP ON NON-ZERO LINK—The content of L is sampled, the next sequential instruction is skipped if L contains a 1. If L contains a 0, the next instruction is executed
SZL	7430	1	10	SKIP ON ZERO LINK—The content of L is sampled, the next sequential instruction is skipped if L contains a 0. If the L contains a 1, the next instruction is executed.
SZA	7440	1	10	SKIP ON ZERO ACCUMULATOR—The content of the AC is sampled; the next sequential instruction is skipped if the AC has all bits which are 0. If any bit in the AC is a 1, the next instruction is executed
SNA	7450	1	10	SKIP ON NON-ZERO ACCUMULATOR—The content of the AC is sampled; the next sequential instruction is skipped if the AC has any bits which are not 0. If every bit in the AC is 0, the next instruction is executed.
SZA SNL	7460	1	10	SKIP ON ZERO ACCUMULATOR, OR SKIP ON NON-ZERO LINK, OR BOTH
SNA SZL	7470	1	10	SKIP ON NON-ZERO ACCUMULATOR AND SKIP ON ZERO LINK
SMA	7500	1	10	SKIP ON MINUS ACCUMULATOR—If the content of AC (0) contains a 1, indicating that the AC contains a negative two's complement number, the next sequential instruction is skipped. If AC (0) contains a 0, the next instruction is executed.
SPA	7510	1	10	SKIP ON POSITIVE ACCUMULATOR—The contents of AC (0) are sampled If AC (0) contains a 0, indicating that the AC contains a positive two's complement number, the next sequential instruction is skipped. If AC (0) contains a 1, the next instruction is executed.
SMA SNL	7520	1	10	SKIP ON MINUS ACCUMULATOR OR SKIP ON NON-ZERO LINK OR BOTH
SPA SZL	7530	1	10	SKIP ON POSITIVE ACCUMULATOR AND SKIP ON ZERO LINK
SMA SZA	7540	1	10	SKIP ON MINUS ACCUMULATOR OR SKIP ON ZERO ACCUMULATOR OR BOTH
SPA SNA	7550	1	10	SKIP ON POSITIVE ACCUMULATOR AND SKIP ON NON-ZERO ACCUMULATOR
SMA SZA SNL	7560	1	10	SKIP ON MINUS ACCUMULATOR OR SKIP ON ZERO ACCUMULATOR OR SKIP ON NON-ZERO LINK OR ALL
SPA SNA SZL	7570	1	10	SKIP ON POSITIVE ACCUMULATOR AND SKIP ON NON-ZERO ACCUMULATOR AND SKIP ON ZERO LINK
CLA	7600	2	10	CLEAR ACCUMULATOR—The AC is loaded with binary 0's
LAS	7604	1,3	15	LOAD ACCUMULATOR WITH SWITCH REGISTER—The content of the AC is loaded with the content of the SR, bit for bit. This is equivalent to a microprogrammed combination of CLA and OSR.
SZA CLA	7640	1,2	10	SKIP ON ZERO ACCUMULATOR THEN CLEAR ACCUMULATOR
SNA CLA	7650	1,2	10	SKIP ON NON-ZERO ACCUMULATOR THEN CLEAR ACCUMULATOR
SMA CLA	7700	1,2	10	SKIP ON MINUS ACCUMULATOR THEN CLEAR ACCUMULATOR
SPA CLA	7710	1,2	10	SKIP ON POSITIVE ACCUMULATOR THEN CLEAR ACCUMULATOR

INSTRUCTION SET (CONTINUED)

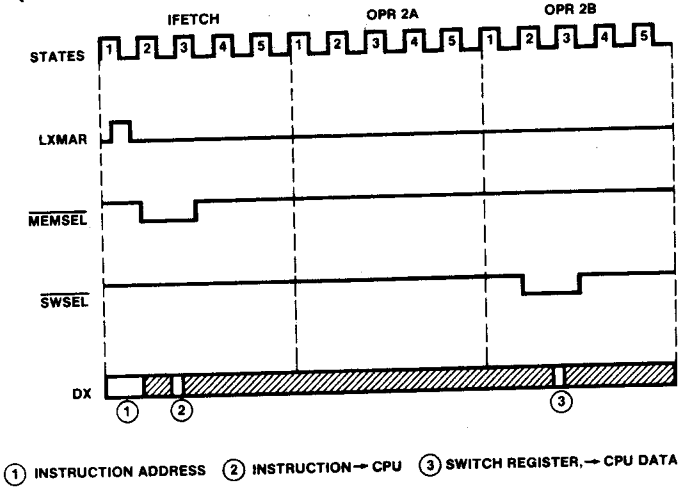


Figure 7: OSR Instruction Timing

GROUP 3 MICROINSTRUCTIONS

Figure 8 shows the instruction format of group 3 microinstructions which requires bits 3 and 11 to contain a 1. Bits 4, 5 or 7 may be set to indicate a specific group 3 microinstruction. If more than one of the bits is set, the instruction is a microprogrammed combination of group 3 microinstructions following the logical sequence listed in Figure 8. All unused bits are "don't care".

0	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	CLA	MQA		MQL				1

*DON'T CARE

LOGICAL SEQUENCES:
1—CLA
2—MQA, MQL
3—ALL OTHERS

Figure 8: Group 3 Microinstruction Format

Table V: Group 3 Operate Microinstructions

MNEMONIC	OCTAL CODE	LOGICAL SEQUENCE	NUMBER OF STATES	OPERATION
NOP	7401	3	10	NO OPERATION—See Group 1 Microinstructions
MQL	7421	2	10	MQ REGISTER LOAD—The content of the AC is loaded into the MQ, the AC is cleared and the original content of the MQ is lost.
MQA	7501	2	10	MQ REGISTER INTO ACCUMULATOR—The content of the MQ is OR'ed with the content of the AC and the result is loaded into the AC. The original content of the AC is lost but the original content of the MQ is retained. This instruction provides the programmer with an inclusive OR operation.
SWP	7521	3	10	SWAP ACCUMULATOR AND MQ REGISTER—The content of the AC and MQ are interchanged accomplishing a microprogrammed combination of MQA and MQL.
CLA	7601	1	10	CLEAR ACCUMULATOR
CAM	7621	3	10	CLEAR ACCUMULATOR AND MQ REGISTER—The content of the AC and MQ are loaded with binary 0's. This is equivalent to a microprogrammed combination of CLA and MQL.
ACL	7701	3	10	CLEAR ACCUMULATOR AND LOAD MQ REGISTER INTO ACCUMULATOR—This is equivalent to a microprogrammed combination of CLA and MQA.
CLA SWP	7721	3	10	CLEAR ACCUMULATOR AND SWAP ACCUMULATOR AND MQ REGISTER—The content of the AC is cleared. The content of the MQ is loaded into the AC and the MQ is cleared.

INSTRUCTION SET (CONTINUED)

INPUT/OUTPUT (IOT) INSTRUCTIONS

The input/output transfer instructions, which have an OP-CODE of 6_8 are used to control the operation of peripheral devices and to transfer data between peripherais and the IM6100. Three types of data transfer may be used to receive or transmit information between the IM6100 and one or more peripheral I/O devices: PROGRAMMED DATA TRANSFER, which provides a straightforward means of communicating with relatively slow I/O devices, such as Teletypes, cassettes, card readers and CRT displays, INTERRUPT TRANSFERS which use the interrupt system to service several peripheral devices simultaneously, and DIRECT MEMORY ACCESS, DMA, which transfers variable-size blocks of data between high-speed peripherals and memory without IM6100 intervention.

IOT INSTRUCTION FORMAT

The Input/Output Transfer Instruction format is represented in Figure 9. The instruction executes in 17 states.

The first three bits, 0-2, are always set to 6_8 (110) to specify an IOT instruction. The low order nine bits are used for device selection and control. PDP-8/e compatible interfaces use bits 3-8 for device selection.and bits 9-11 for control of the selected device. The IM6101 PIE interface uses bits 3-7 for device selection and bits 8-11 for control. In user designed systems, the 512 possible IOT instructions may be alloted according to the user's needs. The nature of this operation for any given IOT instruction depends entirely upon the circuitry designed into the I/O device interface.

PROGRAMMED DATA TRANSFER

Programmed Data Transfer is the easiest, simplest, most convenient and most common means of performing data I/O. For microprocessor applications, it may also be the most cost effective approach. The data transfer begins when the IM6100 fetches an instruction from the memory and recognizes that the current instruction is an IOT (Figure 10). This is referred to as IFETCH and consists of five (5) internal states. The IM6100 sequences the IOT instruction through a 2-cycle execute phase referred to as IOT_A and IOT_B. Bits 0-11 of the IOT instructions are available on DX0-11 at IOT_A • LXMAR; these bits must be latched in an external address register. DEVSEL is active low to enable data transfers between the IM6100 and the peripheral device(s). The selected peripheral device communicates with the IM6100 through 4 control lines - C_0, C_1, C_2 and SKP. In the IM6100 the type of data transfer, during an IOT instruction, is specified by the peripheral device(s) by asserting the control lines as shown in Table VI.

The control line SKP, when low during an IOT, causes the IM6100 to skip the next sequential instruction. This feature is used to sense the status of various signals in the device interface. The C_0, C_1, and C_2 lines are treated independently of the SKP line. In the case of a RELATIVE or ABSOLUTE JUMP, the skip operation is performed after the jump. The input signals to the IM6100, DX 0-11, C_0, C_1, C_2, and SKP, are sampled at IOT_A during DEVSEL • XT_C and the data from the IM6100 is available to the device(s) during that time. IOT_B is used by the IM6100 to perform the operations requested during IOT_A. Both IOT_A and IOT_B consist of six (6) internal states.

In summary, Programmed Data Transfer performs data I/O with a minimum of hardware support. The maximum rate at which programmed data transfers may take place is limited by the IM6100 instruction execution rate, however, the data rate of the most commonly used peripheral devices is much lower than the maximum rate at which programmed transfers can take place in the IM6100. The major drawback associated with Programmed Data Transfer is the IM6100 must hang up in a waiting loop while the I/O device completes the last transfer and prepares for the next transfer. On the other hand, this technique permits easy hardware implementation and simple, economical interface design. For this reason, almost all devices except mass storage units rely on programmed data transfer.

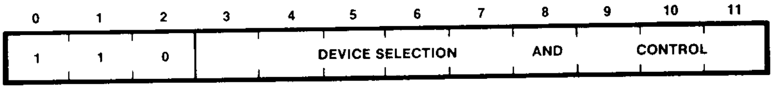


Figure 9: IOT Instruction Format

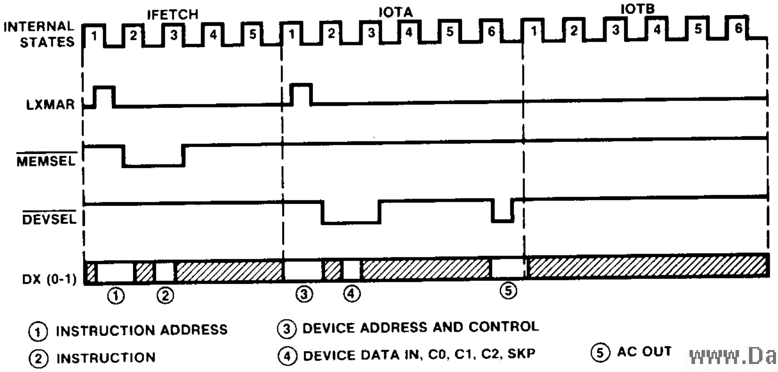


Figure 10: Input-Output Instruction Timing

INSTRUCTION SET (CONTINUED)

Table VI: Programmed I/O Control Lines

CONTROL LINES C_0	C_1	C_2	OPERATION	DESCRIPTION
H	H	H	DEV ←AC	The content of the AC is sent to the device.
L	H	H	DEV ←AC; CLA	The content of the AC is sent to a device and then the AC is cleared.
H	L	H	AC ←AC V DEV	Data is received from a device, OR'ed with the data in the AC and the result is stored in the AC.
L	L	H	AC←DEV	Data is received from a device and loaded into the AC.
*	H	L	PC←PC + DEV	Data from the device is added to the contents of the PC. This is referred to as a RELATIVE JUMP.
*	L	L	PC ←DEV	Data is received from a device and loaded into the PC. This is referred to as an ABSOLUTE JUMP.

*Don't Care

INTERRUPT TRANSFER

PROGRAM INTERRUPT TRANSFERS

The program interrupt system may be used to initiate programmed data transfers in such a way that the time spent waiting for device I/O is greatly reduced or eliminated altogether. This is accomplished by isolating the I/O handling routines from the mainline program and using the interrupt system to ensure that these routines are entered only when an I/O device status is set, indicating that the device is actually ready to perform a data transfer.

The interrupt system allows certain external conditions to interrupt the computer program by driving the INTREQ input Low. If no higher priority requests are outstanding and the interrupt system is enabled, the IM6100 grants the device interrupt at the end of the current instruction. After an interrupt has been granted, the Interrupt Enable Flip-Flop in the IM6100 is reset so that no more interrupts are acknowledged until the interrupt system is re-enabled under program control.

DEVICE INTERRUPT GRANT TIMING

The current contents of the Program Counter, PC, are deposited in location 0000_8 of the memory and the program fetches the instruction from location 0001_8. The return address is available in location 0000_8. This address must be saved in a software stack, before the interrupts are re-enabled, if nested interrupts are permitted. The INTGNT signal, Figure 11, is activated by the IM6100 when a device interrupt is acknowledged; this signal is reset by executing any IOT instruction as shown in Figure 12. The INTGNT signal is necessary to implement an External Vectored Priority Interrupt network. The IM6101 PIE contains the logic necessary to implement both vectored and non-vectored interrupts.

The user program controls the interrupt mechanism of the IM6100 by executing the processor IOT instructions listed in Table VII. Several of these interrupt IOT instructions are also used if the memory is extended beyond 4K words to save and restore extended memory status during interrupt servicing.

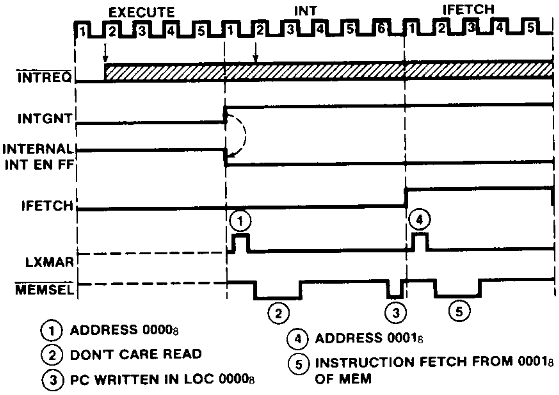


Figure 11: Device Interrupt Grant Timing

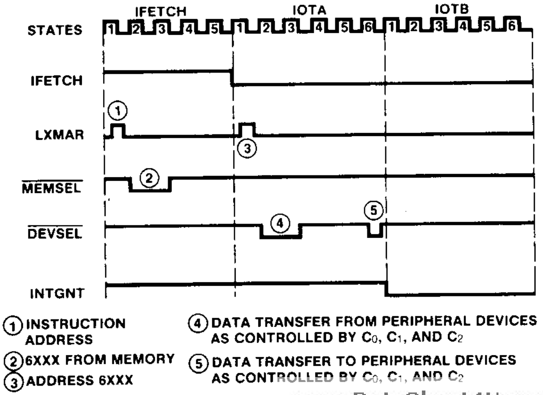
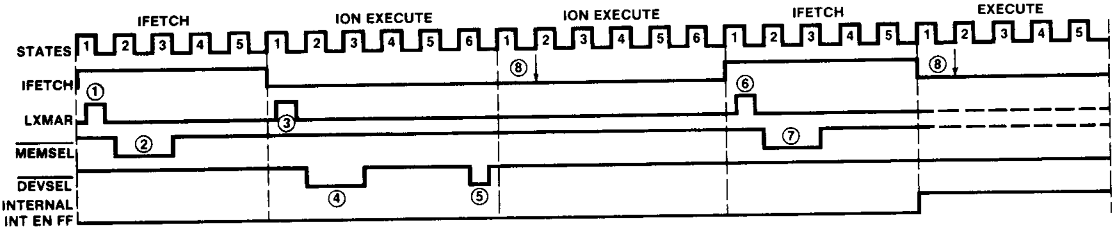

STATES
IFETCH
IOTA
IOTB
1 2 3 4 5 1 2 3 4 5 6 1 2 3 4 5 6
IFETCH
LXMAR
MEMSEL
DEVSEL
INTGNT
① INSTRUCTION ADDRESS
② 6XXX FROM MEMORY
③ ADDRESS 6XXX
④ DATA TRANSFER FROM PERIPHERAL DEVICES AS CONTROLLED BY C0, C1, AND C2
⑤ DATA TRANSFER TO PERIPHERAL DEVICES AS CONTROLLED BY C0, C1, AND C2

Figure 12: Device Interrupt Grant Reset Timing

INSTRUCTION SET (CONTINUED)

Table VII: Processor IOT Instructions

MNE-MONIC	OCTAL CODE	OPERATION
SKON	6000	SKIP IF INTERRUPT ON — If interrupt system is enabled, the next sequential instruction is skipped. The Interrupt system is disabled.
ION	6001	INTERRUPT TURN ON — The internal interrupt acknowledge system is enabled. The interrupt system is enabled after the CPU executes the next sequential instruction. The INTERRUPT ENABLE TIMING is shown in Figure 13.
IOF	6002	INTERRUPT TURN OFF — The interrupt system is disabled. Note that the interrupt system is automatically disabled when the CPU acknowledges an INT request.
SRQ	6003	SKIP IF INT REQUEST — The next sequential instruction is skipped if the INT request bus is low.
GTF	6004	GET FLAGS — The following machine states are read into the indicated bits of AC. bit 0 — Link bit 2 — INT request bus bit 4 — Interrupt Enable FF Other bits may be modified by external devices by controlling the C-lines, (ex. Extended memory control).
RTF	6005	RETURN FLAGS — Link is restored from AC (0). Interrupt system is enabled after the next sequential instruction is executed. All AC bits are available externally to restore external states. (ex. Extended memory control).
SGT	6006	Operation is determined by external devices, if any.
CAF	6007	CLEAR ALL FLAGS — AC and Link are cleared. Interrupt system is disabled.



① INSTRUCTION ADDRESS
② INSTRUCTION FETCH
③ DEVICE ADDRESS (6001$_8$)
④ DON'T CARE DEV READ, SAMPLE C0, C1, C2 & SKP
⑤ DON'T CARE DEV WRITE
⑥ INSTRUCTION ADDRESS
⑦ INSTRUCTION FETCH
⑧ SAMPLE REQUEST LINES

Figure 13: Interrupt Enable FF ON (ION)

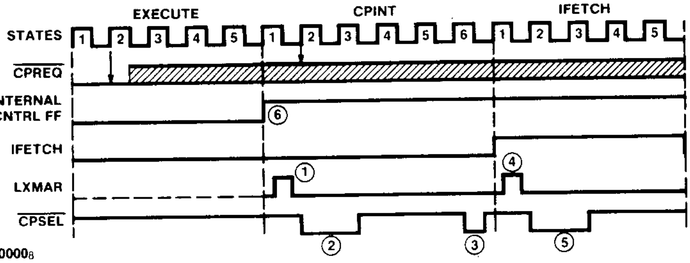
CONTROL PANEL INTERRUPT TRANSFER

The IM6100 supports a memory space completely separate from main memory, called control panel memory. Therefore, the IM6100 control panel and other supervisory functions are implemented in software. This implementation need not use any part of the main memory or change the processor state. This is an important feature, since the final version of the system may not have a control panel and the system designer would like to use the entire capacity of the main memory for the specific system application.

The control panel communicates with the IM6100 with the Control Panel Request, CPREQ, line. The CPREQ is functionally similar to the INTREQ with some important differences. The CPREQ is granted even when the machine is in the HALT state; the IM6100 is temporarily put in the RUN state for the duration of the panel routine. The IM6100 reverts to its original processor state after the panel routine has been executed.

The CPREQ does not affect the interrupt enable system, and the processor IOT instruction, ION is redefined and IOF is ignored while the IM6100 is in the Control Panel Mode. Once a CPREQ is granted, the IM6100 will not recognize any DMAREQ or INTREQ until CPREQ has been fully serviced.

When a CPREQ is granted, the PC is stored in location 0000$_8$ of the Panel Memory and the IM6100 resumes operation at location 7777$_8$. The Panel Memory would be organized with RAM's in the lower pages and PROM's in the higher pages. The control panel service routine would be stored in the higher pages in the nonvolatile PROM's, starting at 7777$_8$.



① ADDRESS 0000$_8$
② DON'T CARE READ
③ PC WRITTEN IN LOC 0000$_8$ OF CP MEM
④ ADDRESS 7777$_8$
⑤ INSTRUCTION FETCHED FROM LOC 7777$_8$ OF CP MEM
⑥ IF CPU IS HALTED, THE RUN IS TRUE AT T1 OF CPINT

Figure 14: Control Panel Interrupt Grant Timing

INSTRUCTION SET (CONTINUED)

A Control Panel Flip-Flop, CNTRL FF, internal to the IM6100, is set when the CPREQ is granted. The CNTRL FF prevents further CPREQ's from being granted.

When the CNTRL FF is set, the Control Panel Memory Select, CPSEL, is active rather than the Memory Select, MEMSEL, for memory references. The CPSEL signal may therefore be used to distinguish the Control Panel Memory from the Main Memory. However, during the Execute phase of indirectly addressed AND, TAD, ISZ or DCA instructions, the MEMSEL is made active. The instructions are always fetched from the control panel memory, and the operand address for indirectly address AND, TAD, ISZ or DCA refers first to the control panel memory for an effective address, which, in turn, refers to a location in the main memory. A main memory location may therefore be examined and changed by indirectly addressed TAD and DCA instructions, Figure 15, respectively. Every location in the main memory is accessible to the control panel routine.

Exiting from the control panel routine is achieved by executing the following sequence with reference made to Figure 16.

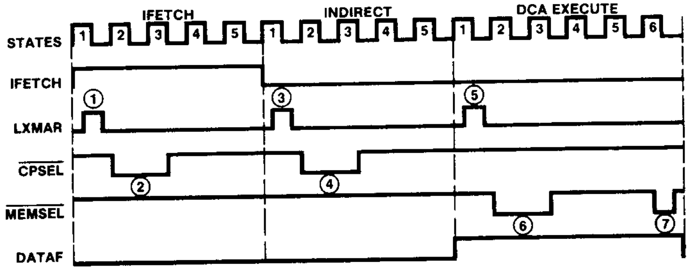
ION
JMP I 0000_8 (Loc 0000_8 in CPMEM)

The ION, 6001_8, instruction will reset the CP FF after executing the next sequential instruction, but will not affect the interrupt system since the CNTRL FF is still active. Location 0000_8 of the CPMEM contains either the original return address, deposited by the IM6100 when the CP routine was entered, or a new starting address defined by the CP routine, for example, by activating the LOAD ADDRESS SWITCH. CPREQ's are normally generated by the manual actuation of the control switches. If the CPU registers must be displayed in real-time, the CPREQ's must be generated by a timer at fixed intervals.

The designer may also make use of the control panel features to implement Bootstrap loaders in the CP Memory so that the loader will be "transparent" to the main memory. Programs will be loaded by DCA I POINTER instruction, the pointer being developed in the CP RAM to point to the main memory location to be loaded.

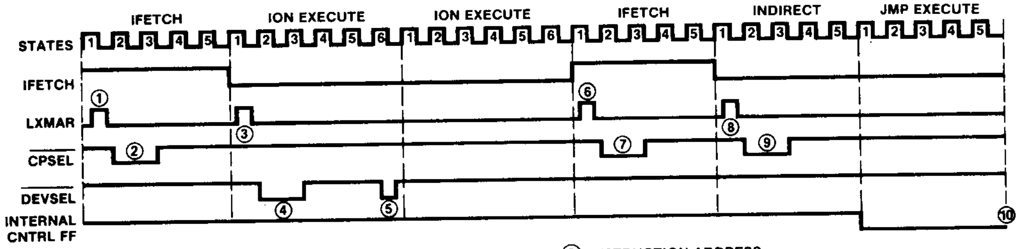
Approximately 64 P/ROM locations are sufficient to implement all the functions of the PDP®-8/e Control Panel. The IM6100 provides for a 12-bit switch register which can be read by the IM6100 under program control with the SWITCH REGISTER, OSR, instruction even without a control panel.

An RTF, 6005_8, instruction also resets the internal CNTRL FF. Exiting from a panel routine can be achieved by activating the RESET line since RESET has a higher priority than CPREQ, see Figure 18. If the RUN/HLT line is pulsed while the IM6100 is in the panel mode, it will 'remember' the pulses(s) but defer any action until the IM6100 exits from the panel mode.



① INSTRUCTION ADDRESS
② INSTRUCTION FROM CP MEMORY
③ EFFECTIVE ADDRESS
④ OPERAND ADDRESS FROM CP MEMORY
⑤ OPERAND ADDRESS
⑥ DON'T CARE MAIN MEM READ
⑦ AC WRITTEN INTO MAIN MEMORY

Figure 15: "DCA Indirect" In Control Panel Routine



① INSTRUCTION ADDRESS
② INSTRUCTION FETCH FROM CP MEM
③ DEVICE ADDRESS (6001_8)
④ DON'T CARE DEVICE READ, SAMPLE C0, C1, C2 & SKP
⑤ DON'T CARE DEVICE WRITE
⑥ INSTRUCTION ADDRESS
⑦ INSTRUCTION FETCH FROM CP MEM
⑧ EFFECTIVE ADDRESS (0000_8)
⑨ JMP ADDRESS FROM CP MEM LOC 0000_8
⑩ IF CPU WAS IN THE HALT STATE, THE RUN IS FALSE AT T1

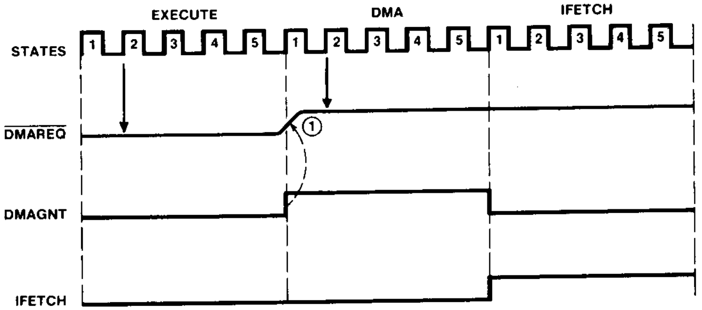
Figure 16: "ION; JMP I 0000_8" In Control Panel Routine

DIRECT MEMORY ACCESS (DMA)

Direct Memory Access, sometimes called data break, is the preferred form of data transfer for use with high-speed storage devices such as magnetic disk or tape units. The DMA mechanism transfers data directly between memory and peripheral devices, and the IM6100 is involved only in setting up the transfer; the transfers take place on a "cycle stealing" basis. The DMA transfer rate is limited only by the bandwidth of the memory and the data transfer characteristics of the device.

The device generates a DMA Request when it is ready to transfer data. The IM6100 grants the DMAREQ by activating the DMAGNT signal at the end of the current instruction as shown in Figure 17. The IM6100 suspends any further instruction fetches until the DMAREQ line is released. The DX lines are tri-stated, all SEL lines are high, and the external timing signals XT_A, XT_B, and XT_C are active and LXMAR remains low. The device which generated the DMAREQ must provide the address and the necessary control signals to the memory for data transfers. The DMAREQ line can also be used as a level sensitive "pause" line.

DMA may also be implemented in a transparent mode without stealing processor cycles by using the DX bus during idle periods. The IM6102 MEDIC operates in this manner.



① DMAREQ REMOVED AFTER DMAGNT

Figure 17: Direct Memory Access (DMA)

INTERNAL PRIORITY STRUCTURE

After an instruction is completely sequenced, the major state generator scans the internal priority network as shown in Figure 18. The state of the priority network decides the next sequence of the IM6100.

The request lines, RESET, CPREQ, RUN/HLT, DMAREQ and INTREQ, are sampled in the last cycle of an instruction execution, at time T1. The worst case response time of the IM6100 to an external request is, therefore, the time required to execute the longest instruction preceded by any 6-state execution cycle. For the IM6100, this is an autoindexed ISZ, 22 states, preceded by any 6-state execution cycle instruction.

When the IM6100 is initially powered up, the state of the timing generator is undefined. The generator is automatically initialized with a maximum of 34 clock pulses. The request inputs, as the IM6100 is powered on, must span at least 58 clock pulses to be recognized, 34 clocks for the counter to initialize and a maximum of two IM6100 cycles (20 to 24 clocks) for the state generator to sample the request lines. A positive transition on RUN/HALT should occur at least 10 clock pulses after RESET for it to be recognized.

The internal priority is RESET, CPREQ, RUN/HLT, DMAREQ, INTREQ, and IFETCH.

IFETCH

If no external requests are pending, the IM6100 fetches the next instruction pointed to by the contents of the PC. The IFETCH line is active during the cycle in which the instruction is fetched. External devices can monitor DX, 0-2, during IFETCH-XT_A to determine the functional class of the current instruction. For example, the external memory extension hardware must know when JMP or JMS instructions are fetched to implement the Extended Memory Control. The IM6102 does this to implement extended memory addressing.

The Programmable Logic Array, PLA, in the IM6100 sequences the IM6100 to execute the fetched instruction. All INDIRECT and AUTOINDEX Memory Reference Instructions go through a common state sequence to generate the Effective Address, EA, of the operand. The subsequent sequence, referred to as the EXECUTE phase, is controlled by the functional class of the instruction. The EXECUTE phase of AND, TAD, DCA, JMS, JMP and OPR Group 3 Microinstructions consists of only one cycle. ISZ and IOT have a 2-cycle EXECUTE phase. OPR Group 1 and Group 2 Microinstructions have an optional second cycle, depending on the microcoding of the OPR instructions. An IM6100 cycle consists of 5 states, T_1, T_2, T_3, T_4 and T_5, with an optional sixth state, T_6, for Output Transfers (WRITE).

The state sequence for internal (processor) and external IOT instructions are identical. The Device Address and Control bits are available in the External Address Register for internal IOT instructions. External hardware, for example Extended Memory Control, can control the C-lines for data transfers to implement Get Flags (GTF), Return Flags (RTF), and Clear All Flags (CAF) instructions. External Control of the C-lines is necessary to implement these internal IOT instructions since the flag bits may be distributed both inside and outside the IM6100.

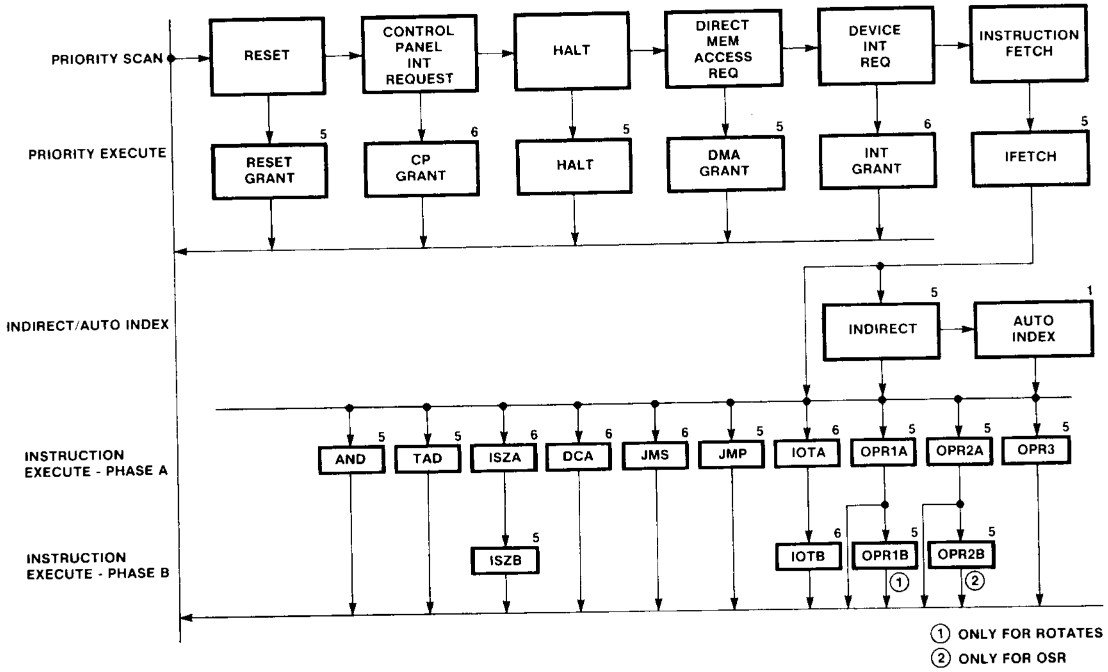


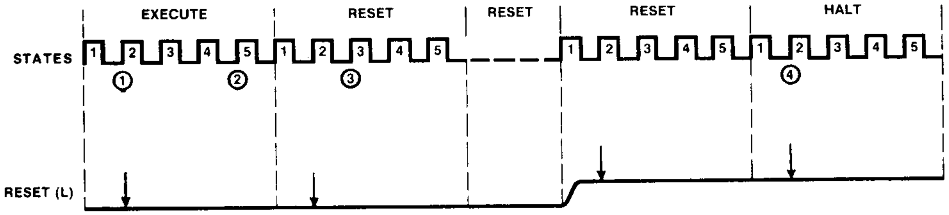
Figure 18: Major Processor States and Number of Clock Cycles in Each State

8

RESET

The Reset initializes all internal IM6100 flags and clears the AC and the LINK. The machine is halted.

As long as the RESET line is low, the IM6100 remains in the reset state and the DX lines are three stated. The IM6100 continues to provide the external timing signals XT_A, XT_B and XT_C, all SEL lines are high, and the PC is set to 7777_8. In most applications, the higher memory locations utilize P/ROM's or ROM's. Therefore, a power-up routine starting at the highest memory location can be used to initialize the system. It is also possible to force entry into control panel memory on power-up.



① REQUESTS SAMPLED AT T1 OF THE FINAL EXECUTE PHASE
② EXECUTE MAY BE 5/6 STATES
③ PC IS SET TO 7777_8
④ CPU HALTS

Figure 19: Reset Timing

RUN/HALT

RUN/HLT changes the state of the IM6100's RUN/HLT flip-flop. Pulsing the line low causes the IM6100 to alternately run and halt. The RUN/HLT line is normally high. The IM6100 recognizes the positive transition of the signal.

The RUN/HLT flip-flop can be put in the halt state under program control by executing the HLT, 7402_8, instruction. When the IM6100 is halted, RUN/HLT is functionally identical to the CONTINUE switch of the PDP-8/e control panel and the RUN signal is low. The RUN signal can be used to power down external circuitry for a low power system.

The RUN/HLT can also be used to make the IM6100 execute one instruction at a time as shown in Figure 21. The RUN/HLT combines the functional features of STOP, CONTINUE, and SINGLE INSTRUCTION as defined by the PDP-8/e Control Panel.

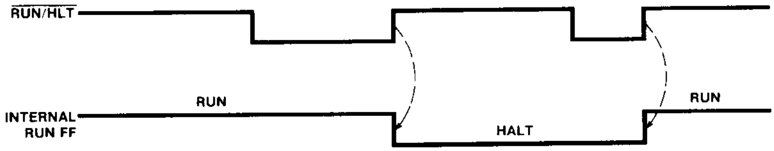
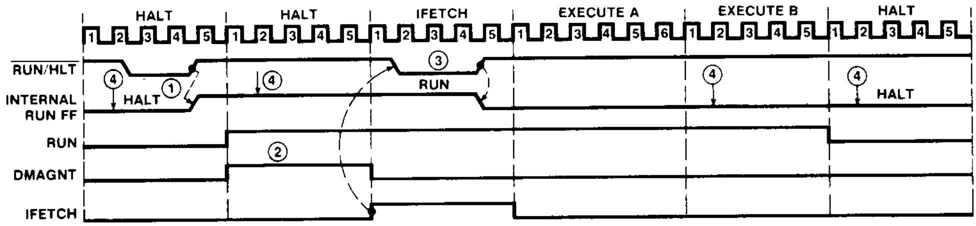


Figure 20: Run/Halt Timing



① RUN/HLT PULSE FOR "SINGLE STEP"
② DMAGNT ON FOR 1 CYCLE FOR HALT TO RUN TRANSITION
③ TRIGGER RUN/HLT WITH IFETCH
④ RUN FF SAMPLED IN THE LAST EXECUTE CYCLE

Figure 21: "Single Step" With Run/Hlt

WAIT

The IM6100 samples the WAIT line during input-output data transfers (Figure 22). The WAIT line, if low, controls the transfer duration. If WAIT is active during input transfers (READ), the CPU waits in the T2 state. For an output transfer (WRITE), WAIT controls the time for which the write data is maintained on the DX lines by extending the T6 state. The wait duration is an integral multiple of the oscillator time period — 250nsec at 4MHz.

The WAIT mechanism is an ideal way of providing for slower memory and peripheral devices in the system without significant degradation in system performance. For example, if one waits for all reads and writes for one delay unit (250nsec at 4MHz), the system throughput is reduced by less than 3%.

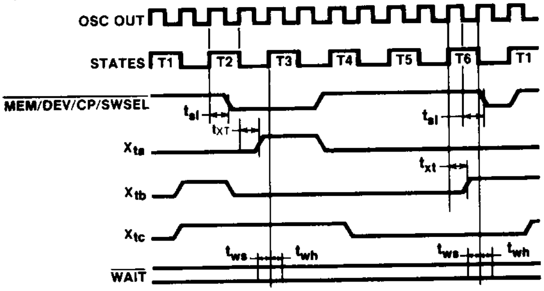


Figure 22: Wait Line Sampling Timing

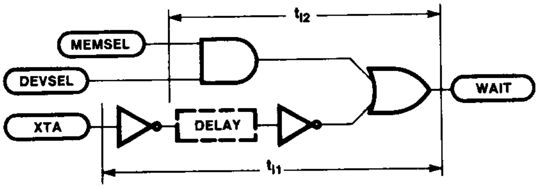


Figure 23: Memory And Input Transfer Wait Circuit

The circuit shown in Figure 23 will make the IM6100 wait during main memory and device input (READ) transfers. MEMSEL or DEVSEL, being low, will assert WAIT low. When XTA becomes active high, the WAIT line is asserted high after a delay. The wait duration is controlled by the delay in the XTA-WAIT path (t_{L1}).

The following conditions must be satisfied to obtain x units of delay during READ's:

$$t_{sl(max)} + t_{l2(max)} + t_{ws} < t_s$$

$$t_{xt(min)} + t_{l1(min)} - t_{wh} \geq x \frac{t_s}{2}$$

$$t_{xt(max)} + t_{l1(max)} + t_{ws} < (x + 1) \frac{t_s}{2}$$

For example, for an IM6100 I device operating at 4MHz, 5.0V and 25° C, the constraints to be met to obtain 1 unit of delay (250nsec) are as follows:

$t_{l2(max)} < t_s - t_{sl(max)} - t_{ws}$

$< 500 - 300 - 30$

< 170nsec

$t_{l1(min)} \geq \frac{t_s}{2} - t_{xt(min)} + t_{wh}$

$\geq 250 - 100 + 30$

≥ 180nsec

$t_{l1(max)} < t_s - t_{xt(max)} - t_{ws}$

$< 500 - 250 - 30$

< 220nsec

Note that the delay circuit can be as simple as an R-C network in conjunction with CMOS logic. Note also that the WAIT can be made selective on main memory, device, control panel memory or switch register select line.

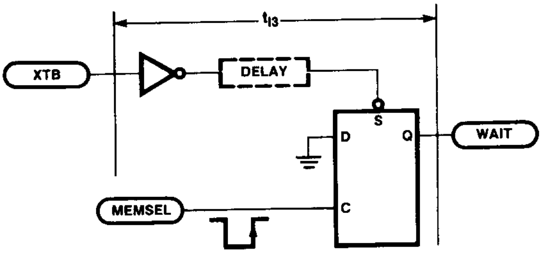


Figure 24: Write Transfer Wait Circuit

Figure 24 shows a logic implementation to wait during WRITE's only.

The rising edge of MEMSEL (or CPSEL or DEVSEL) during READ clocks in a zero on the WAIT line. XTB, after a delay, releases the WAIT line. Every WRITE pulse is preceded by a READ pulse, and if no write operation is performed in a cycle, the T6 state is not entered and the WAIT line is not sampled.

For x units of delay, the following conditions must be met:

$$t_{xt(min)} + t_{l3(min)} - t_{wh} \geq x \frac{t_s}{2} \text{ and}$$

$$t_{xt(max)} + t_{l3(max)} + t_{ws} < (x + 1) \frac{t_s}{2}$$

In the circuit shown in Figure 25, the WAIT signal is normally asserted low and it is released by XTA during READ's and XTB during WRITE's. Note that WAIT is active for all data transfers. Since XTA and XTB have identical timing relative to the WAIT sample point, the constraints to be satisfied are as follows:

$$t_{xt(min)} + t_{l4(min)} - t_{wh} \geq x \frac{t_s}{2} \text{ and}$$

$$t_{xt(max)} + t_{l4(max)} + t_{ws} < (x + 1) \frac{t_s}{2}$$

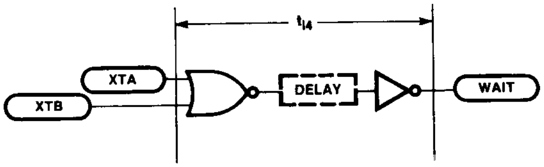


Figure 25: Data Transfer Wait Circuit