FINAL

Am4601

Programmable-Flags, 512 x 9 FIFO

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- 512 x 9 RAM-based FIFO
- 25 and 35 ns access times
- Two fixed flags; full and empty
- Two programmable flags; programmable from 1 to 511
- Programmable polarity for all four flags
- Data, $\overline{R}$, and $\overline{W}$ pinout compatible with industry-standard (720X) FIFOs
- Programmable depth mode

GENERAL DESCRIPTION

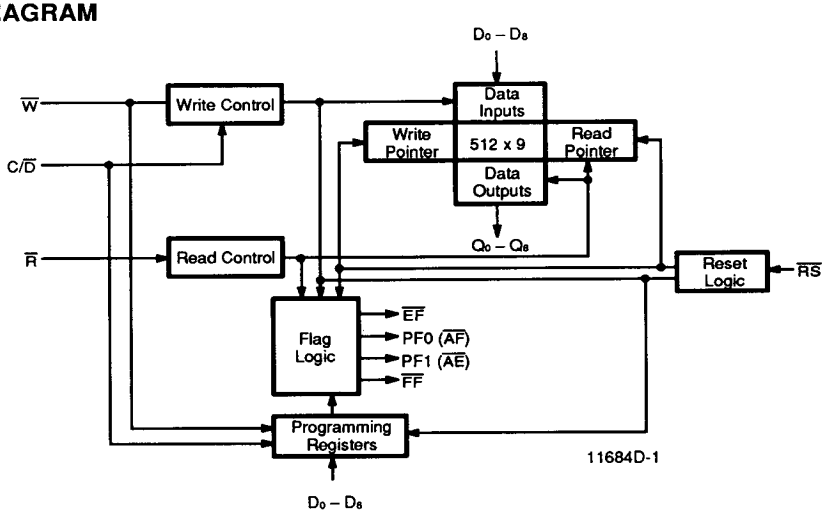
The Am4601 is a RAM-based CMOS FIFO that is 512 words deep with 9-bit wide words. It has two fixed flags, Full and Empty, and two programmable flags, programmable from 1 to 511 in increments of one.

The Am4601 can accept data and output data asynchronously and simultaneously at data rates from 0 to 28.5 MHz. The Am4601 has four programming registers which allow the user to program the two programmable flags, to program the polarity of each of the four flags, and to change the total usable depth of the FIFO.

The Am4601 data, $\overline{R}$ and $\overline{W}$ pinout is identical to industry standards 720X FIFOs.

The Am4601 is ideally suited for data buffering applications such as in communication, image processing, mass storage, DSP and printing systems. The programmable flags and the programmable depth mode of the Am4601 are especially useful for advance status signaling and message packing required for high performance systems.

BLOCK DIAGRAM



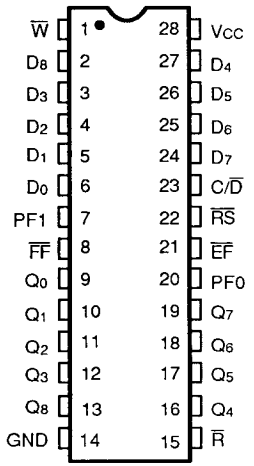
PRODUCT SELECTOR GUIDE

Part Number	Am4601-25	Am4601-35
Access Time	25 ns	35 ns
Maximum Power Supply Current	90 mA	80 mA
Operating Frequency	28.5 MHz	22.2 MHz
Operating Range	Com'l	Com'l

Publication# 11684 Rev. D Amendment/0
Issue Date: September 1992

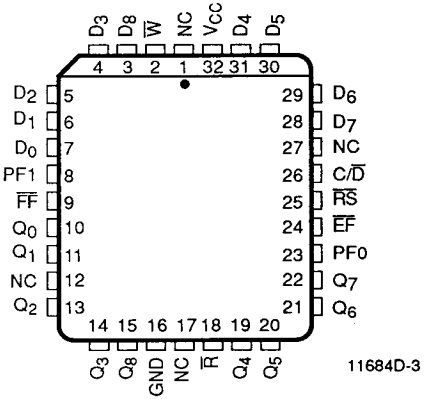
CONNECTION DIAGRAMS
Top View

DIP



11684D-2

PLCC

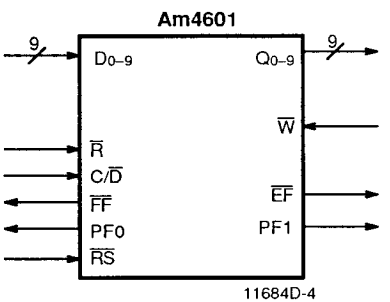


11684D-3

Notes:

Pin 1 is marked for orientation for plastic packages.
NC: No Connection.

LOGIC SYMBOL



11684D-4

PIN DESCRIPTION

D_8–D_0

9-Bit Input Data

Q_8–Q_0

9-Bit Output Data

$\overline{R}$

Active-Low Read Control

$\overline{W}$

Active-Low Write Control

C/$\overline{D}$

Command/Data Selection

C/$\overline{D}$ =1: Enables writing to the programming registers (FIFO is empty and PB = 0)

C/$\overline{D}$ = 0: Enables writing to the FIFO memory.

$\overline{RS}$

Master Reset

Resets, when low, the FIFO address pointers, and initializes the programming registers to their default values.

$\overline{EF}$

Empty Flag

Programmable polarity

$\overline{FF}$

Full Flag

Programmable polarity

PF0

Programmable Flag 0

Defaults to Almost Full ($\overline{AF}$) upon reset.

PF1

Programmable Flag 1

Defaults to Almost Empty ($\overline{AE}$) upon reset.

V_{CC}

Power Supply Pin

Input, +5 Volts

GND

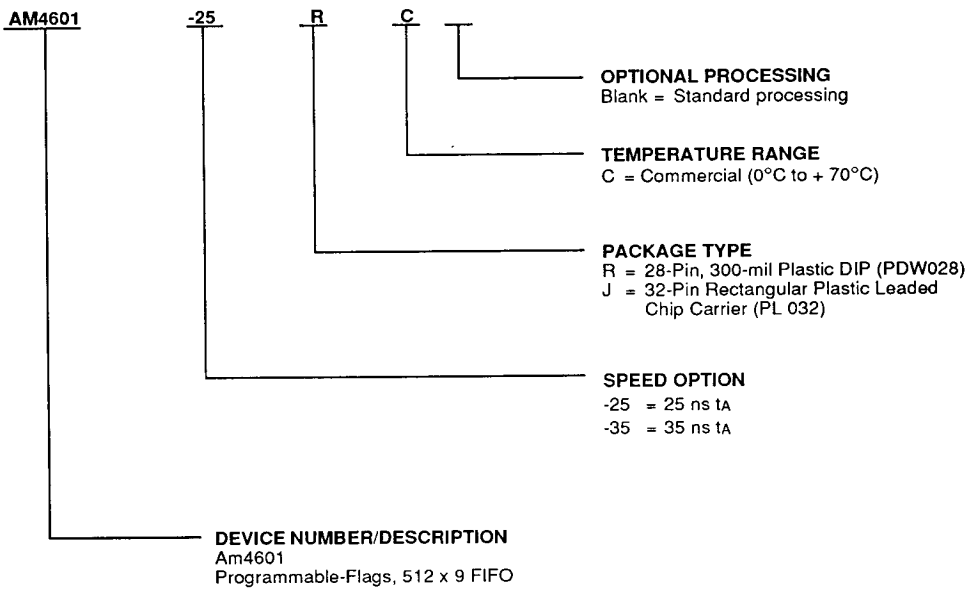
Ground Supply Pin

Input, 0 Volts

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM4601-25	RC, JC
AM4601-35	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, and to check on newly released valid combinations.

FUNCTIONAL DESCRIPTION

The Am4601 Programmable-Flags CMOS FIFO is designed around a 512 x 9 dual-port static RAM array (see block diagram). Two dedicated address counters – read address pointer and write address pointer-facilitate the sequential FIFO operation. The address pointers roll over to address zero after reaching address 511.

Flag logic determines the difference between the Write and Read address pointers and generates the four status flags. The trip points of two programmable flags and the polarity of all four flags are programmed via the programming registers. The flag logic also prevents overwriting the FIFO when it is full and reading the FIFO when it is empty.

Resetting the FIFO is realized by pulsing the $\overline{RS}$ pin. Both address pointers are initialized to zero and the programming registers return to their default vales.

For applications that require a FIFO of a different depth than 512, the Am4601 can be programmed to block writing when the programmable flag PF0 is asserted. Thus the Am4601 functions as a programmable-depth FIFO.

Resetting The FIFO

Upon power up, the FIFO must be initialized with a Reset cycle (See Figure 7). For a valid Reset cycle to occur, both the Read ($\overline{R}$) and Write ($\overline{W}$) must be HIGH through the low-to-high transition of the Reset ($\overline{RS}$) signal. The Reset cycle initializes the FIFO to an empty condition and the programming registers to a default state. Both address pointers are reset to zero, the programmable depth mode is disabled, and the Register Programming bit in register 3 (PB) is enabled to allow programming of the registers. The polarity of the Full flag, the Empty flag and programmable flag 0 is set to assertive-low, and the polarity of programmable flag 1 is set to assertive-high. PF0 is initialized to trigger at address 496 ($\overline{AF}$) and PF1 is set to trigger at 16 ($\overline{AE}$).

Read / Write Operations

Am4601 read and write operations are controlled by $\overline{R}$, $\overline{W}$, and C/$\overline{D}$ control lines. $\overline{R}$ controls the read operation, $\overline{W}$ controls the write operation, and C/$\overline{D}$ controls the data into the FIFO memory or into the programming registers.

The falling edge of $\overline{R}$ initiates the read cycle and valid data appears on the outputs (Q_0–Q_8) after the access time (t_A). Data remains valid until t_{dv} after the rising edge of $\overline{R}$ and then the outputs go to high-impedance state. Cycling the $\overline{R}$ also increments the Read address pointer. When the FIFO is empty, $\overline{R}$ is ignored.

Data may be written into the FIFO memory (C/$\overline{D}$ = 0)or into the programming registers (C/$\overline{D}$ = 1, the FIFO is empty, and PB = 0). $\overline{W}$ going low initiates the write cycle. Data on the input must be stable t_{DS} before and t_{DH} after the rising edge of $\overline{W}$.

When C/$\overline{D}$ = 1, PB = 0, and the FIFO is empty, the data will be written into the programming registers. Otherwise the data will be written into the FIFO memory, and the write address counter will be incremented. When the FIFO is full, or when the write blocking bit is set and the FIFO reaches the PF0 programmed limit, the $\overline{W}$ signal is ignored.

A special case of write occurs when a write operation is initiated while the part is full. The next read will cause $\overline{FF}$ to go inactive, and data can then be latched into the FIFO t_{WPF} after the rising edge of $\overline{FF}$ (see Figure 11).

A special case of read occurs when a read operation is initiated while the part is empty. The data latched in by the next write will be accessed t_A ns after the rising edge of $\overline{EF}$. Read is held active, and cannot be deasserted until t_{RPE} after the rising edge of $\overline{EF}$ (see Figure 10).

Flag Operation

After a Reset cycle the FIFO is empty and the flags reflect the FIFO's empty status and the flag defaults ($\overline{EF}$ = 0, $\overline{FF}$ = 1, PF0 = 1, PF1 = 0). After the first write operation, the Empty flag changes its state (to "not empty") following the rising edge of $\overline{W}$. PF0 and PF1 are asserted following the falling edge of the $\overline{W}$ during the cycle that makes the number of words in the FIFO equal to the flag's programmed values. The Full flag changes its state (from "not full" to "full") after the falling edge of $\overline{W}$ at the cycle that writes into the last location of the FIFO (see Figure 8).

When reading the first word out of a full FIFO, the Full flag changes its state (from "full" to "not full") after the rising edge of $\overline{R}$. PF0 and PF1 change their state following the rising edge of the $\overline{R}$ during the cycle that makes the FIFO hold fewer words than the programmed value. When reading the last word from the FIFO the Empty flag changes status from "not-empty" to "empty" following the falling edge of $\overline{R}$ (See Figure 9).

When asserted, the Full flag prevents any further write operations into the FIFO. Similarly, the Empty flag, when asserted, prevents any further read operations. When the FIFO is programmed as a programmable depth FIFO (Write Blocking Bit = 1), PF0, when asserted, prevents any further write operations into the FIFO. Thus, PF0 serves in this mode as a Full flag.

PF0 and PF1 assertion is similar to the Full Flag. They are asserted following the falling edge of the Write which takes the FIFO to the programmed depth. They are deasserted on the rising edge of the Read which empties the FIFO to the programmed depth. PF0 and PF1 can be programmed to any value from 1 to 511. The value 0 is illegal for either PF0 or PF1.

During programming, the flags may change their states. The flags must be ignored during the programming sequence.The flags settle to their proper states following the fourth register write at the end of the programming sequence.

Programming the Flags

The Am4601 has four, 8-bit wide, programming registers that allow the programming of:

- Programmable flag 0 (PF0) value
- Programmable flag 1 (PF1) value
- Write blocking bit (WB)
- Programming blocking bit (PB)

The four programming registers may be programmed after master reset or when the FIFO is empty and PB = 0. The registers are programmed by four sequential write operations with C/$\overline{D}$ = 1 (see Figures 5 & 6).

C/$\overline{D}$ = 1 routes the data from D_0– D_7 inputs into the programming registers and prevents the FIFO's write address pointer from incrementing. A special register pointer increments automatically after each programming write operation and points to the next register. Figure 1 shows the data format of the programming registers.

The four registers must be programmed in four sequential write operation. Stopping before the completion of four programming cycles will result in FIFO malfunction. The flags change their states during programming and must be ignored during these four cycles. The programming registers are write-only.

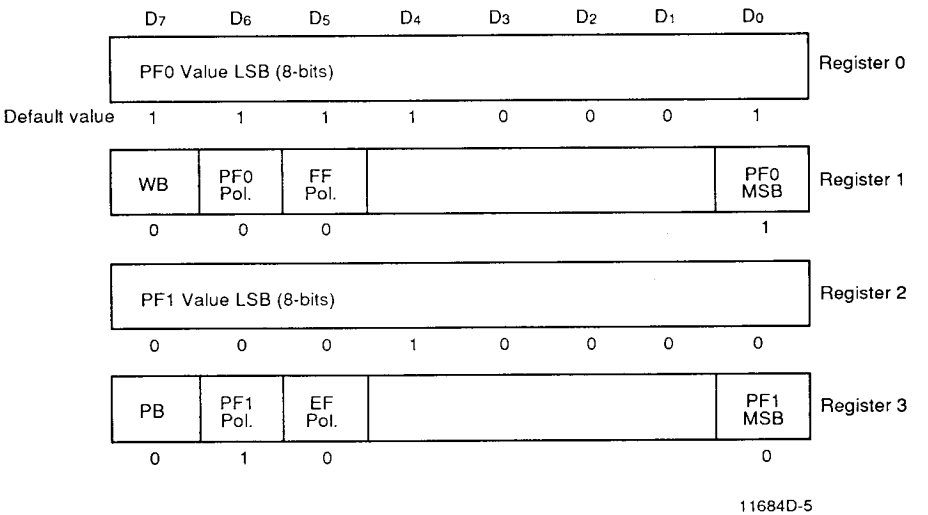


Figure 1. Programming Registers Format and Defaults

Register 0:

Stores the 8 LSB bits of Programmable flag 0 (PF0) location. A Reset cycle sets register 0 to the value 496 (16 words from full).

Register 1:

Stores the MSB location-bit of PF0 and the following control bits:

WB

Write Blocking control bit; enables write blocking with PF0.

WB = 1: Changes the PF0 flag to act as a FULL flag. The PF0 flag will block the write pulses to the FIFO after the flag is set. This feature can be used in applications where only a part of the 512 bytes is needed. In those applications the programmable FIFO is used as a programmable depth FIFO.

WB = 0: The PF0 flag has no control on the write pulses to the FIFO. The write pulses will be blocked only after the FULL flag is set.

After Reset, WB bit is 0.

PF0 Pol.

Selects the polarity of the PF0 programmable flag.

PF0 Pol = 0: The programmable flag will be asserted-low.

PF0 Pol = 1: The programmable flag will be asserted-high.

After Reset, PF0 Pol bit is 0.

FF Pol.

Selects the polarity of the FULL flag.

$\overline{FF}$ Pol. = 0: The FULL flag will be active-low.

$\overline{FF}$ Pol. = 1: The FULL flag will be active-high.

After Reset, $\overline{FF}$ Pol. bit is 0.

Register 2:

Stores the 8 LSB bits of Programmable flag 1 (PF1) location. After Reset PF1 is set to 16 (16 words from empty).

Register 3:

Stores the MSB bit of PF1 location and the following control bits:

PB

Programming Blocking Bit; enables blocking of programming.

PB = 0: The registers are programmed by writing to the FIFO while the C/$\overline{D}$ input is 1, while the FIFO is empty. With C/$\overline{D}$ = 1, writes to the registers while the FIFO is not empty will be addressed to the FIFO core array and not the registers. The registers can be accessed again after a master reset, or when the FIFO is empty.

After a reset cycle the PB bit is 0.

PB = 1: PB = 1 disables any further writes to the registers. The programming registers cannot be written into until PB is reset to 0 by a Master reset. PB is set by writing into the register and is reset by a reset cycle.

PF1 Pol.

Selects the polarity of the PF1 programmable flag.

PF1 Pol. = 0: The programmable flag will be asserted-high.

PF1 Pol. = 1: The programmable flag will be asserted-low.

After Reset, PF1 Pol. bit is 1.

EF Pol.

Selects the polarity of the EMPTY flag.

EF Pol. = 0: The EMPTY flag will be active-low.

EF Pol. = 1: The EMPTY flag will be active-high.

After reset, the EF Pol. bit is 0.

AFTER MASTER RESET

After master-reset the programming registers go to the following default values:

Register 0	1 1 1 1	0 0 0 1
Register 1	0 0 0 X	X X X 1
Register 2	0 0 0 1	0 0 0 0
Register 3	0 1 0 X	X X X 0

PF0: Is set at 497, asserted-low to act as an Almost Full, active-low flag ($\overline{AF}$). When Low, PF0 will indicate that the FIFO has more than 496 words.

PF1: Is set at 16, asserted-high to act as an Almost Empty, active-low flag ($\overline{AE}$). When Low, PF1 will indicate that the FIFO has less than 16 words in it.

Empty flag Polarity: Active-low

Full flag Polarity: Active-low

WB: The write block is reset to 0; i.e. the FIFO will be in normal operation mode.

PB: The programming block is reset to 0 to allow programming of the registers.

APPLICATIONS

The Am4601 programmable FIFO provides high speed data buffering for digital systems. The FIFO's programmable flags allow the system designer to optimize data throughput and to achieve better system performance. The most common uses of the programmable flags are advance system signaling of boundary conditions, and message packing. The programmable depth mode is useful to buffer complete messages of different lengths.

Programmable Flags for Advance System Signaling

In many systems data is generated or received by a subsystem with slow response time (The system may generate or receive data at one rate, but requires substantially longer time to stop, or start the data stream). For such a system, time must be allowed so that the system can react to the full or empty flags and prevent data loss due to writing into a full FIFO (or reading out of an empty FIFO). In many cases, the user uses the half full flag (HF) of a standard FIFO as a system level signal to stop and start the data flow. While this solution works in many applications, it has some drawbacks. The FIFO cannot be filled more than half way, resulting in inefficient use of a deeper (and costlier) FIFO where a less expensive, half-depth FIFO could be used. In other cases, the system is slowed to accommodate the time it takes to react to the flags. There, the system and the FIFO are not used to their best performance.

The Am4601 programmable flags are specifically designed to answer this application. The user may program the two programmable flags to trip at any point. Thus, if the FIFO is getting near to full condition, a programmable flag can be set to trigger at a desired number of cycles before the FIFO fills up. Thus, the system gets an early warning with enough time to react to the boundary condition flags (full and empty). At the same time, the FIFO and the system may run at full speed allowed by the data stream.

Programmable Flags for Message Packing

In other systems, throughput and performance are achieved by sending complete messages through the system data channels. The reason is that a substantial overhead is associated with establishing a communication channel. Be it a simple parallel bus, or a sophisticated communication trunk, it takes a long time for the arbitration circuits to grant a subsystem access to the communication channel. Passing a short, or partial message through the channel awards a small benefit for the hefty expense of arbitrating for the channel. A much more efficient use of communications channels is to send complete messages of preferred sizes.

When a FIFO is used at either side of a communication channel, the system may benefit from writing into the FIFO a complete message, or reading a complete message from it. Since a standard FIFO provides only empty, full, and half-full status flags, the system designer is limited to using the FIFO for message lengths that fit into the FIFO status signals. Moreover, the FIFO can be used to contain one or two messages only.

Using the Am4601 programmable flags, the user can signal the system that space is provided to receive a message of a preferred length. Also, the flags may be used to tell the system on the either side that the FIFO contains at least one complete message for transmission. This way, the system throughput may be optimized using complete message transmission and reception.

Programmable Depth

Where a long string of data of a particular length needs to be received or transmitted, the Am4601 can have its depth altered. The Am4601 is then programmed to disallow further write operation once one of the programmable flags (PF0) is triggered. The FIFO depth can be therefore programmed to any length between 1 and 511. In this mode the full flag is replaced by the programmable flag.

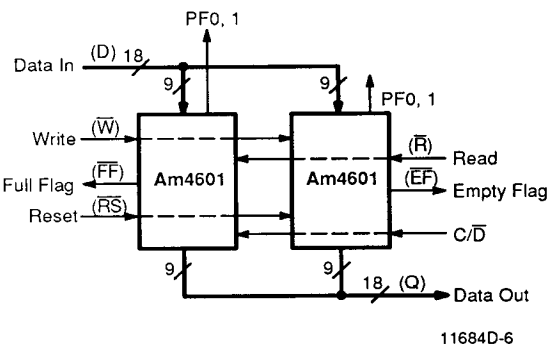


Figure 2. Width-Expansion to Form a 512 x 18 FIFO

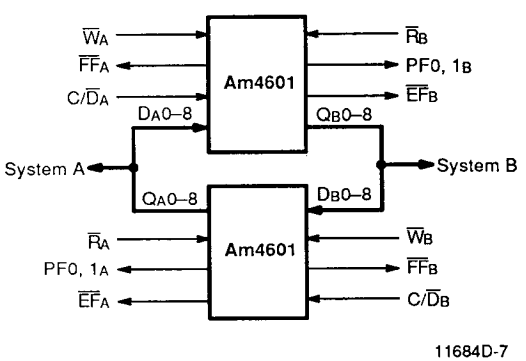


Figure 3. Bidirectional FIFO Mode

ABSOLUTE MAXIMUM RATINGS

Supply Voltage ..... –0.5 V to +7.0 V
All Signal Voltages ..... –0.5 V to +7.0 V
DC Output Current ..... 20 mA
Power Dissipation ..... 1.0 W
Operating Temperature ..... –55°C to +125°C
Storage Temperature ..... –65°C to +155°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Supply Voltage ..... +4.5 V to +5.5 V
Operating Temperature ..... 0°C to +70°C

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Commercial		Unit
			Min.	Max.	
I_{OH}	Output High Current	V_{OH} = 2.4 V, V_{CC} = 4.5 V	–2.0		mA
I_{OL}	Output Low Current	V_{OL} = 0.4 V, V_{CC} = 4.5 V	+6		mA
V_{IH}	Input High Voltage	(Note 1)	2.0	V_{CC} + 0.5	V
V_{IL}	Input Low Voltage	(Note 1)		0.8	V
I_{IX}	Input Leakage Current	GND ≤ V_{IN} ≤ V_{CC}, V_{CC} = 5.5 V	–10	10	µA
I_{OZ}	Output Leakage Current	GND ≤ V_{OUT} ≤ V_{CC}, V_{CC} = 5.5 V	–10	10	µA
I_{CC1}	Static Operating Supply Current	GND ≤ V_{OUT} ≤ V_{CC}, V_{CC} = 5.5 V (Note 2)		15	mA
I_{CC2}	Dynamic Operating Current, (22.2 MHz Max.)	GND ≤ V_{OUT} ≤ V_{CC}, V_{CC} = 5.5 V (Note 2)		80	mA
I_{CC3}	Dynamic Operating Current, (28.5 MHz Max.)	GND ≤ V_{OUT} ≤ V_{CC}, V_{CC} = 5.5 V (Note 2)		90	mA

Notes:

1. V_{IL} and V_{IH} are input conditions of output tests and are not themselves directly tested. V_{IL} and V_{IH} are absolute voltages with respect to device ground and include all overshoots due to system and/or tester noise. Do no attempt to test these values without suitable equipment.
2. I_{CC} measurements are made with outputs open.

CAPACITANCE (Note 3) (V_{CC} = 5.0, T_A = 25°C, f = 1.0 MHz)

Parameter Symbol	Parameter Descriptions	Test Conditions	Typ.	Unit
C_{IN}	Input Capacitance	V_{IN} = 0 V	5	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0 V	7	

Note:

3. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range unless otherwise specified

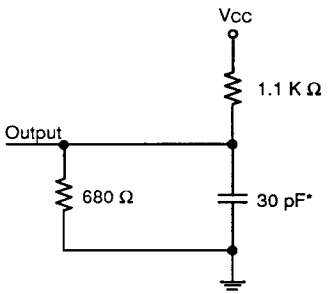
No.	Parameter Symbol	Parameter Description	Am4601-25		Am4601-35		Unit
			Min.	Max.	Min.	Max.	
Write Timing							
1	t_{WC}	Write Cycle Time	35		45		ns
2	t_{WPW}	Write Pulse Width	15		15		ns
3	t_{WR}	Write Recovery Time	8		10		ns
4	t_{DS}	Data Setup Time	15		15		ns
5	t_{DH}	Data Hold Time	2		2		ns
6	t_{AS}	C/$\overline{D}$ Setup Time	5		5		ns
7	t_{AH}	C/$\overline{D}$ Hold Time	0		0		ns
8	t_{WFF}	Write LOW to Full Flag LOW/HIGH		30		30	ns
9	t_{WPF}	Write LOW to PF0, 1 asserted		45		45	ns
10	t_{WEF}	Write HIGH to Empty Flag HIGH/LOW		30		30	ns
11	t_{WLZ}	Write HIGH to data bus at LOW-Z (Note 1)	10		10		ns
Read Timing							
12	t_{RC}	Read Cycle Time	35		45		ns
13	t_{A}	Access Time		25		35	ns
14	t_{RR}	Read Recovery Time	8		10		ns
15	t_{RPW}	Read Pulse Width	25		35		ns
16	t_{RLZ}	Read LOW to data bus at LOW-Z (Note 1)	5		5		ns
17	t_{DV}	Data Valid from Read HIGH	5		5		ns
18	t_{RHZ}	Read HIGH to data bus HIGH-Z (Note 1)		18		20	ns
19	t_{RFF}	Read HIGH to Full Flag HIGH/LOW		30		30	ns
20	t_{RPF}	Read HIGH to PF0, 1 asserted		45		45	ns
21	t_{REF}	Read LOW to Empty Flag LOW/HIGH		30		30	ns
Reset Timing							
22	t_{RSC}	Reset Cycle Time		35	45		ns
23	t_{RS}	Reset Pulse Width	25		25		ns
24	t_{RSS}	Reset Setup Time	25		35		ns
25	t_{RSR}	Reset Recovery Time	10		10		ns
26	t_{EFL}	Reset to Empty and PF1 Flags LOW		35		45	ns
27	t_{FFH}	Reset to Full and PF0 Flags HIGH		35		45	ns

Note:

1. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where these parameters may be affected.

AC TEST CONDITIONS

Input pulse levels	GND to 3.0 V
Input rise and fall times	5 ns
Input timing reference levels	1.5 V
Output timing reference levels	1.5 V
Output loads	See Figure 4



11684D-8

* Includes jig and scope capacitances

Figure 4. AC Test Load

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must Be Steady	Will Be Steady
	May Change from H to L	Will Be Changing from H to L
	May Change from L to H	Will Be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High Impedance "Off" State

KS000010

SWITCHING WAVEFORMS

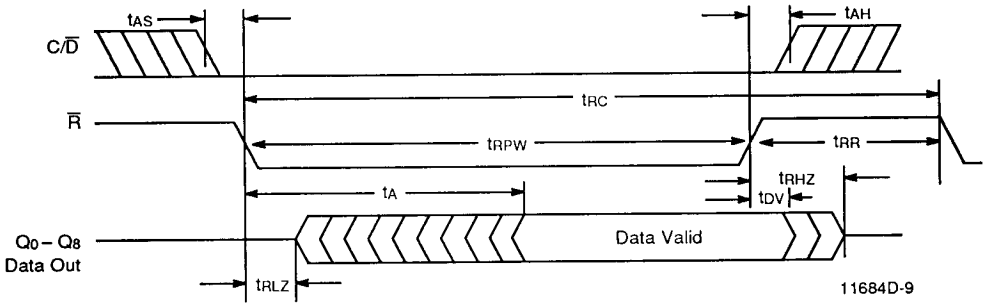


Figure 5. Read Timing Diagram

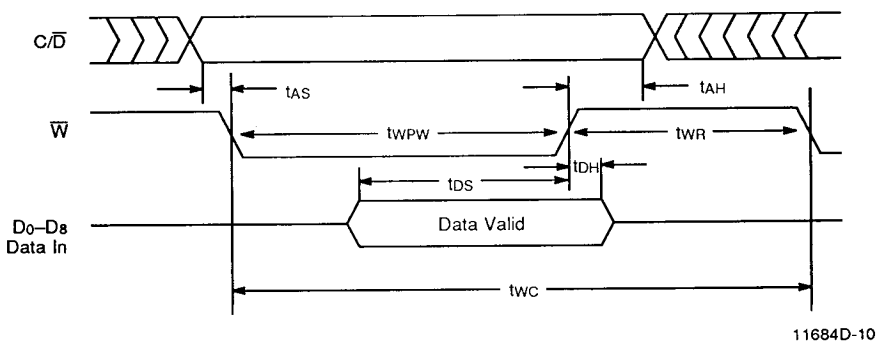


Figure 6. Write Timing Diagram

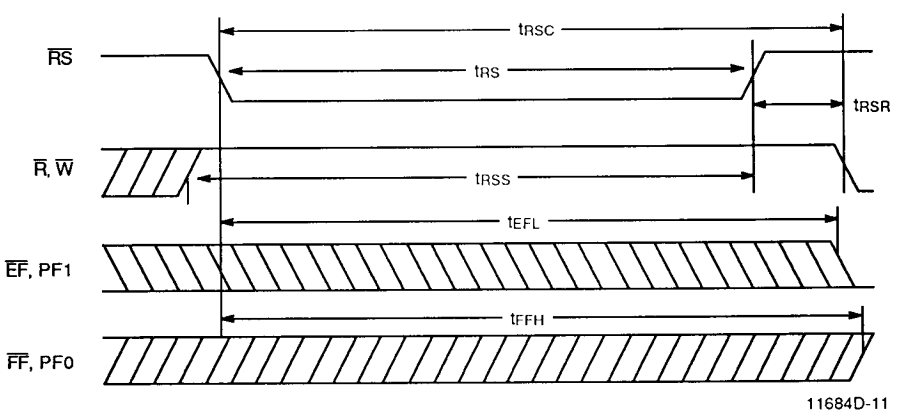


Figure 7. Reset Timing Diagram

SWITCHING WAVEFORMS

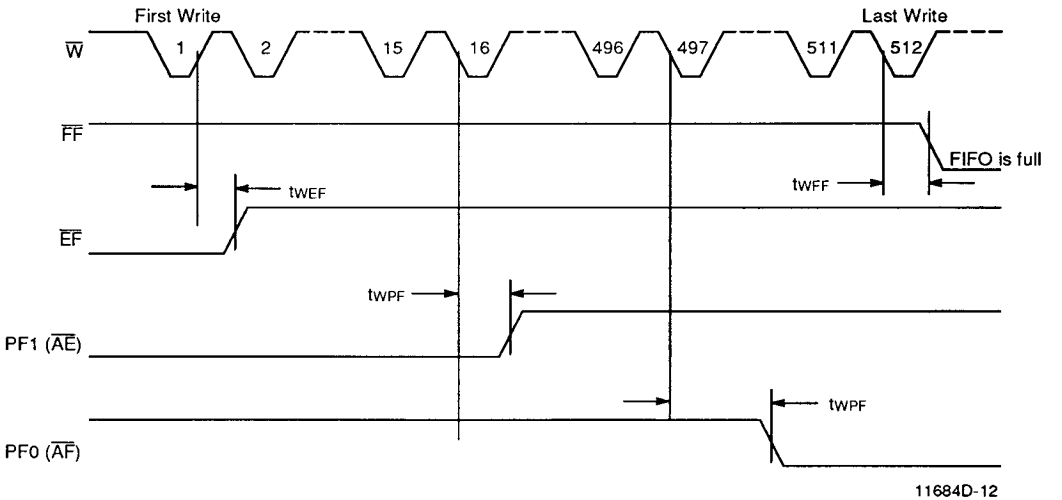


Figure 8. Flag Timing (Write Cycles) – Operated in Default Value

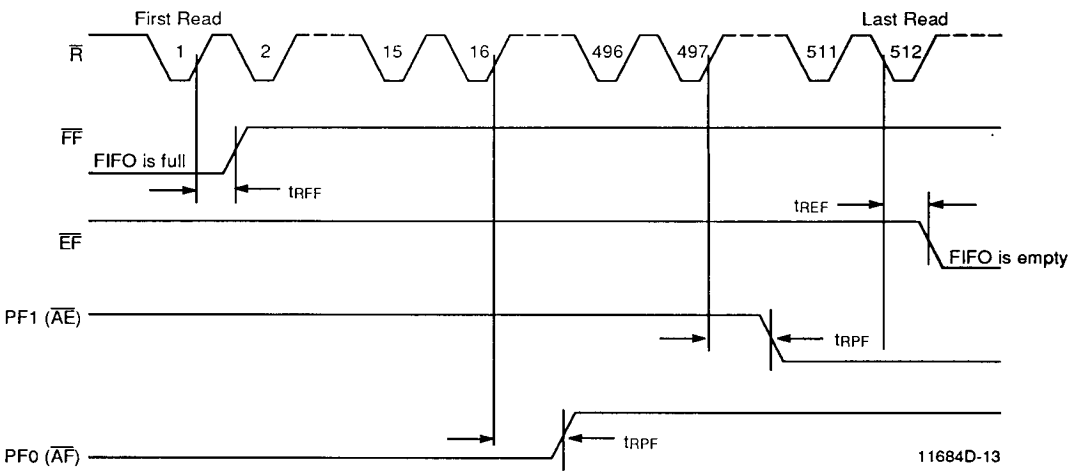
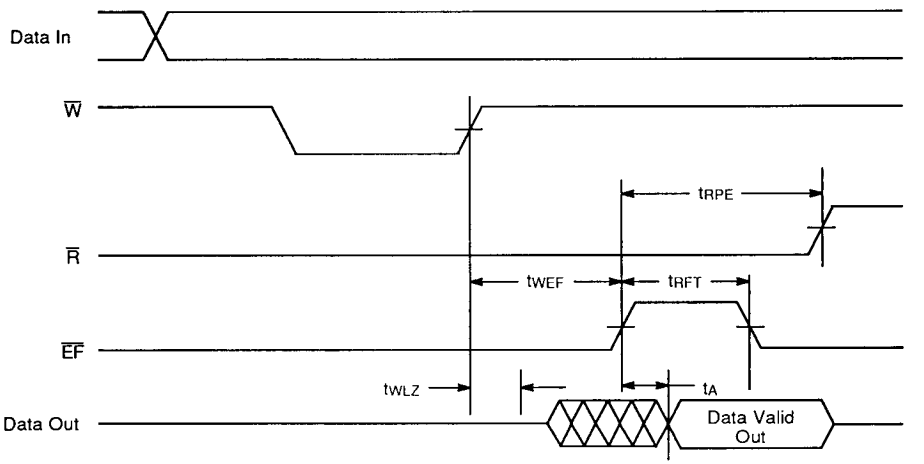
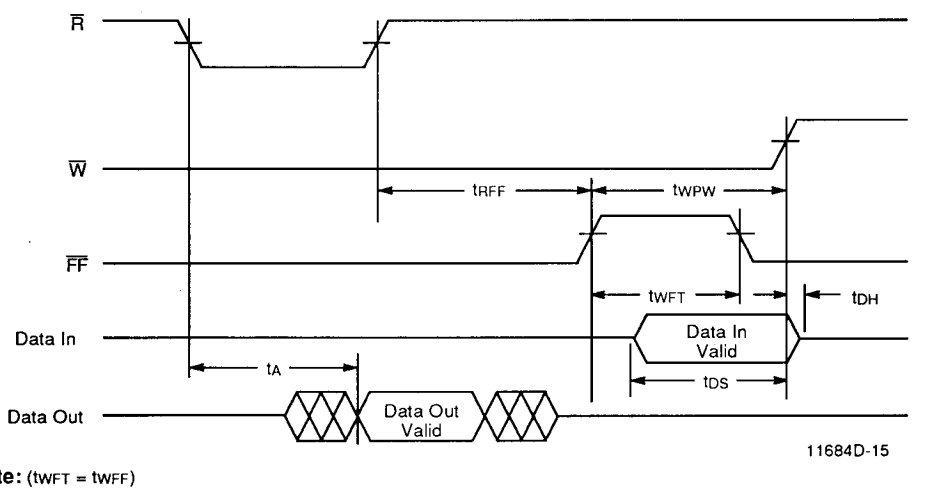


Figure 9. Flag Timing (Read Cycles) – Operated in Default Value



Note: ($t_{RPE} = t_{RPW}$, $t_{RFT} = t_{REF}$)

Figure 10. Read Data Flow-Through Mode



Note: ($t_{WFT} = t_{WFF}$)

Figure 11. Write Data Flow-Through Mode