OKI Semiconductor

This version: Jan. 1999
Previous version: Mar. 1998

MSM7630

Universal Speech Processor

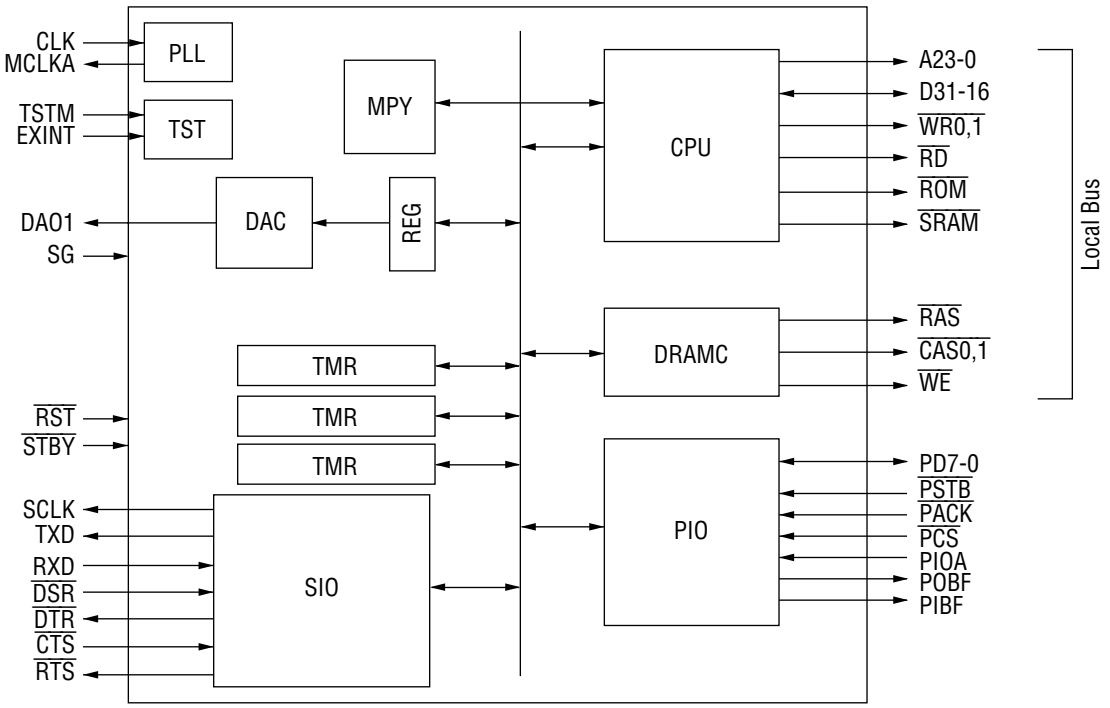
GENERAL DESCRIPTION

The MSM7630 is a speech processor LSI device with internal D/A converter. It is optimized for speech output applications such as text-to-speech conversion.

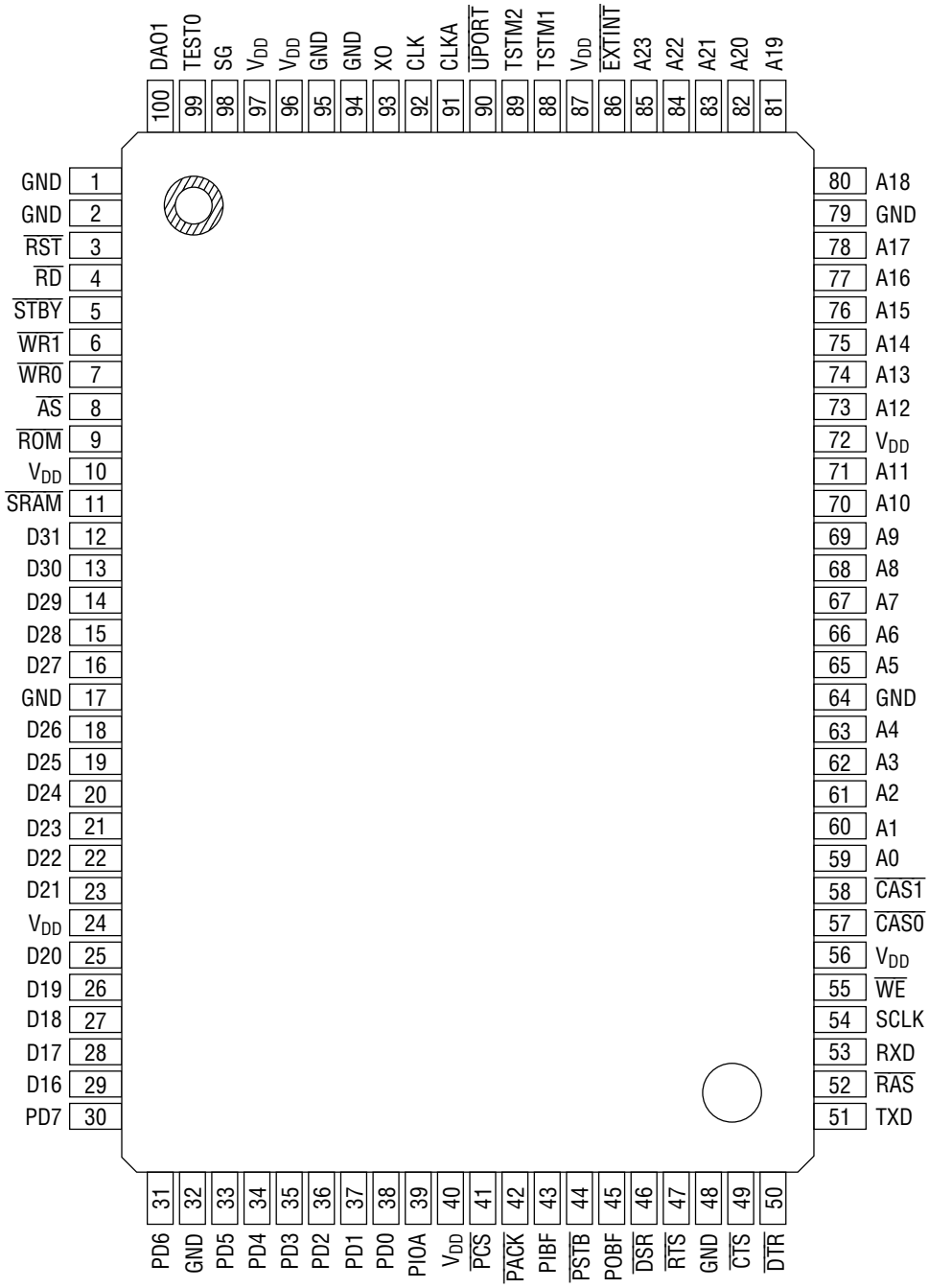
FEATURES

- Parallel and serial interfaces
- Single 3.3V power supply
- 5V interface available
- Internal 16-bit x 16-bit to 32-bit multiplier (2-clock data throughput)
- 26 VAX MIPS performance at 40 MHz operation (when using ordinary ROM/SRAM)
- Package: 100-pin plastic QFP (QFP100-P-1420-0.65-BK) (Product name: MSM7630GS-BK)

BLOCK DIAGRAM



PIN CONFIGURATION (TOP VIEW)



100-Pin Plastic QFP

PIN DESCRIPTIONS

Symbol	Type	Description
D31-16	I/O	16-bit data bus. 8-bit devices are accessed through D31-24.
A23-0	O	24-bit address bus. DRAM addresses are output from A13-0.
$\overline{ROM}$	O	ROM select signal. $\overline{ROM}$ indicates that ROM space is assigned to the specified address. It is used as a chip select signal.
$\overline{SRAM}$	O	SRAM select signal. $\overline{SRAM}$ indicates that SRAM space is assigned to the specified address. It is used as a chip select signal.
$\overline{RD}$	O	Read signal. $\overline{RD}$ is active during both 8-bit and 16-bit reads.
$\overline{WR0,1}$	O	Write signals. $\overline{WR0}$ corresponds to writes from D31-24, and $\overline{WR1}$ corresponds to writes from D23-16.
$\overline{RAS}$	O	Row address strobe. $\overline{RAS}$ is active during both 8-bit and 16-bit reads.
$\overline{CAS0,1}$	O	Column address strobe. $\overline{CAS0}$ corresponds to accesses from D31-24, and $\overline{CAS1}$ corresponds to accesses from D23-16.
$\overline{WE}$	O	Write enable. $\overline{WE}$ is active during writes to DRAM space as the DRAM write signal.
$\overline{AS}$	O	Address strobe.
TXD	O	Serial data output.
RXD	I	Serial data input.
$\overline{DTR}$	O	Control signal indicating SIO can transmit and receive.
$\overline{DSR}$	I	Input signal indicating that modem is in operable state.
$\overline{RTS}$	O	SIO transmit request signal.
$\overline{CTS}$	I	Input signal indicating that modem can transmit.
SCLK	O	Synchronous transfer clock output.
PD7-0	I/O	Parallel port data input/output.
$\overline{PACK}$	I	Parallel port read signal. Set high for Centronics interface.
$\overline{PSTB}$	I	Parallel port write signal. Strobe signal for Centronics interface.
$\overline{PCS}$	I	Parallel port chip select signal.
PIOA	I	Parallel port address signal. Selects data or status during an access.
POBF	3-state	Output port buffer full. Indicates that data has been set in the output buffer.
PIBF	3-state	Input port buffer full. Indicates that there is data in the input buffer. Busy output signal for Centronics interface.
$\overline{UPORT}$	O	General flag output signal.
TEST0	I	Connects with SG.
DAO1	O	D/A converter output.
SG	I	Signal ground. Connects with TEST0.
CLK	I	Clock input signal.
XO	O	Clock signal. Inverse of CLK.
CLKA	O	Internal clock signal.
$\overline{RST}$	I	Reset input.
$\overline{STBY}$	I	Standby signal. $\overline{STBY}$ suspends operation and places the MSM7630 in a standby state.
$\overline{EXTINT}$	I	External interrupt signal.
TSTM2,1	I	Test mode select input signal.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Condition	Rating	Unit
Power Supply Voltage	V_{DD}	Ta = 25°C (excluding TEST0)	–0.3 to +4.5	V
Input Voltage	V_{IN}	Ta = 25°C	–0.3 to +5.5	V
Storage Temperature	T_{STG}	—	–55 to +125	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Conditon	Range	Unit
Power Supply Voltage	V_{DD}	—	3.0 to 3.6	V
Operating Temperature	T_{op}	—	–40 to +85	°C

ELECTRICAL CHARACTERISTICS

DC Characteristics

(V_{DD} = 3.0 to 3.6 V, Ta = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
"H" Input Voltage	V_{IH}	Excluding CLK	2.2	—	—	V
"L" Input Voltage	V_{IL}	Excluding CLK	—	—	0.8	V
"H" Input Voltage	V_{IH}	CLK	$0.8 \times V_{DD}$	—	—	V
"L" Input Voltage	V_{IL}	CLK	—	—	$0.2 \times V_{DD}$	V
"H" Output Voltage	V_{OH}	I_{OH} = –4 mA	2.4	—	—	V
"L" Output Voltage	V_{OL}	I_{OL} = 4 mA	—	—	0.4	V
Input Leakage Current	I_{LI}	$0 \le V_{IN} \le V_{DD}$	–10	—	+10	µA
Output Leakage Current	I_{LO}	$0 \le V_{OUT} \le V_{DD}$	–10	—	+10	µA
Dynamic Supply Current	I_{DO}	V_{DD} = 3.6 V, f_{OPE} = 20 MHz	—	—	120	mA
Static Supply Current	I_{DS}	—	—	—	1.5	mA
D/A Output Relative Accuracy	$\lvert V_{DAE} \rvert$	No load	—	—	10	mV
D/A Output Impedance	R_{DA}	—	12	20	28	kΩ

AC Characteristics

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Source Oscillation Period	t_{OSC}	—	25	—	50	ns
Input Clock Low-Level Minimum Width	t_{W_CLKL}	—	13	—	—	ns
Input Clock High-Level Minimum Width	t_{W_CLKH}	—	8	—	—	ns
Operating Period	t_{CYC}	—	25	—	50	ns
CLKA Delay Time	t_{CLK}	—	—	—	12	ns
XO Delay Time	t_{XO}	—	—	—	7	ns
Required $\overline{RST}$ Time	t_{W_RST}	—	1024	—	—	t_{CYC}
A Delay Time	t_A	—	—	—	22	ns
D Setup Time	t_{S_D}	—	2	—	—	ns
D Hold Time	t_{H_D}	—	6	—	—	ns
D Delay Time	t_D	—	—	—	25	ns
$\overline{RD}$ Delay Time	t_{RD}	—	—	—	20	ns
$\overline{WR}$ Delay Time	t_{WR}	Falling	—	—	22	ns
		Rising	—	—	22 + 0.5 t_{CYC}	ns
UPORT Delay Time	t_{UPORT}	—	—	—	20	ns
EXINT Setup Time	t_{S_EXINT}	—	2	—	—	ns

ROM, SRAM Access

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{RD}$ Pulse Width	t_{W_RD}	ROM, SRAM 3τ to 12τ Access	2	—	11	t_{CYC}
$\overline{WR}$ Pulse Width	t_{W_WR}	SRAM 3τ to 12τ Access	1.5	—	10.5	t_{CYC}
A to $\overline{RD}$ Time	t_{W_ARD}	ROM, SRAM 3τ, 4τ Access	—	1	—	t_{CYC}
		ROM, SRAM 5τ to 12τ Access	—	2	—	t_{CYC}
A to $\overline{WR}$ Time	t_{W_AWR}	SRAM 3τ to 12τ Access	—	1	—	t_{CYC}
$\overline{WR}$ to $\overline{SRAM}$ Time	t_{W_WRSRAM}	SRAM 3τ to 12τ Access	—	1	—	t_{CYC}
$\overline{ROM}$ Delay Time	t_{ROM}	—	—	—	20 + 0.5 t_{CYC}	ns
$\overline{SRAM}$ Delay Time	t_{SRAM}	—	—	—	20 + 0.5 t_{CYC}	ns
$\overline{ROM}$ Pulse Width	t_{W_ROM}	ROM 3τ to 12τ Access	3	—	12	t_{CYC}
$\overline{SRAM}$ Pulse Width	t_{W_SRAM}	SRAM 3τ to 12τ Access	3	—	12	t_{CYC}
WR to D Time	t_{W_WRD}	SRAM 3τ Access	—	0	—	t_{CYC}
		SRAM 4τ to 12τ Access	—	1	—	t_{CYC}

DRAM Access

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{RAS}$ Delay Time	t_{RAS}	—	—	—	18	ns
$\overline{RAS}$ Pulse Width	t_{W_RAS}	—	3	—	Note 1	t_{CYC}
A to $\overline{RAS}$ Time	t_{W_ARAS}	—	1	—	—	t_{CYC}
$\overline{CAS}$ Delay Time	t_{CAS}	2nτ access falling edge	—	—	18 + 0.5 t_{CYC}	ns
		Normal	—	—	18	ns
$\overline{CAS}$ Pulse Width	t_{W_CAS}	Normal	1.5	—	2	t_{CYC}
		Refresh	4	—	5	t_{CYC}
A to $\overline{CAS}$ Time	t_{W_ACAS}	—	0.5	—	1	ns
$\overline{RAS}$ to $\overline{CAS}$ Time	t_{W_RASCAS}	—	1.5	—	2	t_{CYC}
$\overline{WE}$ to $\overline{CAS}$ Time	t_{W_WECAS}	—	1.5	—	2	t_{CYC}
$\overline{WE}$ Delay Time	t_{WE}	—	—	—	20	ns
$\overline{WE}$ Pulse Width	t_{W_WE}	—	3	—	Note 1	t_{CYC}
A to $\overline{WE}$ Time	t_{W_AWE}	—	—	1	—	t_{CYC}
Required Precharge Time	t_{W_PREC}	—	1	—	Note 2	t_{CYC}
$\overline{CAS}$ to $\overline{RAS}$ Time	t_{W_CASRAS}	—	—	1	—	t_{CYC}
$\overline{CAS}$ to D Time	t_{W_EDO}	Hyper Mode	—	—	1	t_{CYC}

General Device Access

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{AS}$ Delay Time	t_{AS}	—	—	—	18	ns

When DS bit = 0

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{AS}$ Pulse Width	t_{W_AS}	4τ to 7τ Access (X bit = 0)	2	—	5	t_{CYC}
		8τ to 14τ Access (X bit = 1)	6	—	12	t_{CYC}
A to $\overline{AS}$ Time	t_{W_AAS}	4τ to 7τ Access (X bit = 0)	—	1	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	1	—	t_{CYC}
$\overline{RD}$ Pulse Width	t_{W_RD}	4τ to 7τ Access (X bit = 0)	2	—	5	t_{CYC}
		8τ to 14τ Access (X bit = 1)	6	—	12	t_{CYC}
A to $\overline{RD}$ Time	t_{W_ARD}	4τ to 7τ Access (X bit = 0)	—	1	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	1	—	t_{CYC}
$\overline{WR}$ Pulse Width	t_{W_WR}	4τ to 7τ Access (X bit = 0)	2	—	5	t_{CYC}
		8τ to 14τ Access (X bit = 1)	6	—	12	t_{CYC}
A to $\overline{WR}$ Time	t_{W_AWR}	4τ to 7τ Access (X bit = 0)	—	1	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	1	—	t_{CYC}
D to WR Time	t_{W_DWR}	4τ to 7τ Access (X bit = 0)	—	0	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	0	—	t_{CYC}

When DS bit = 1

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{AS}$ Pulse Width Note 3	t_{W_AS}	4τ to 7τ Access (X bit = 0)	2	—	5	t_{CYC}
		8τ to 14τ Access (X bit = 1)	6	—	12	t_{CYC}
A to $\overline{AS}$ Time	t_{W_AAS}	4τ to 7τ Access (X bit = 0)	—	1	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	1	—	t_{CYC}
$\overline{RD}$ Pulse Width	t_{W_RD}	4τ to 7τ Access (X bit = 0)	2	—	5	t_{CYC}
		8τ to 14τ Access (X bit = 1)	6	—	12	t_{CYC}
A to $\overline{RD}$ Time	t_{W_ARD}	4τ to 7τ Access (X bit = 0)	—	1	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	1	—	t_{CYC}
$\overline{WR}$ Pulse Width	t_{W_WR}	4τ to 7τ Access (X bit = 0)	2	—	5	t_{CYC}
		8τ to 14τ Access (X bit = 1)	6	—	12	t_{CYC}
A to $\overline{WR}$ Time	t_{W_AWR}	4τ to 7τ Access (X bit = 0)	—	2	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	3	—	t_{CYC}
D to WR Time	t_{W_DWR}	4τ to 7τ Access (X bit = 0)	—	1	—	t_{CYC}
		8τ to 14τ Access (X bit = 1)	—	2	—	t_{CYC}

Serial Interface

(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{RTS}$ Delay Time	t_{RTS}	—	—	—	20	ns
Required RXD Time	t_{W_RXD}	—	1/bps	—	—	s
RXD Setup Time	t_{S_RXD}	—	0.5/bps	—	—	s
RXD Hold Time	t_{H_RXD}	—	0.5/bps	—	—	s
$\overline{CTS}$ Setup Time	t_{S_CTS}	—	0	—	—	ns
$\overline{CTS}$ Hold Time	t_{H_CTS}	—	0	—	—	ns
TXD Delay Time	t_{TXD}	—	—	—	20	ns
TXD Pulse Width	t_{W_TXD}	—	1/bps	—	—	s
$\overline{DTR}$ Delay Time	t_{DTR}	—	—	—	20	ns
SCLK Delay Time	t_{SCLK}	—	—	—	20	ns
SCLK Pulse Width	t_{W_SCLK}	—	1/bps	—	—	s

Parallel Interface

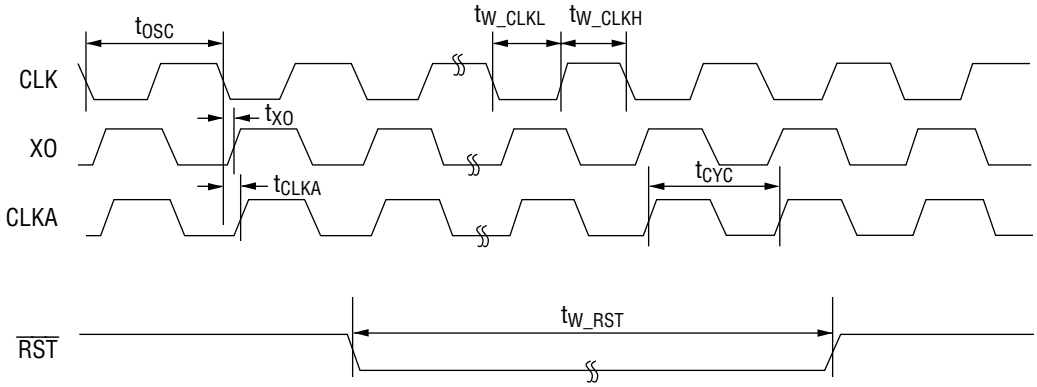
(V_{DD} = 3.0 to 3.6 V, T_{OPE} = –40 to +85°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
$\overline{PACK}$ to PD Delay Time	t_{PACK}	—	—	—	22	ns
$\overline{PACK}$ to PD Hi Z Delay Time	t_{PRDZ}	—	—	—	22	ns
$\overline{PCS}$ Setup Time for $\overline{PSTB}$/$\overline{PACK}$	t_{S_PCS}	—	0	—	—	ns
$\overline{PCS}$ Hold Time for $\overline{PSTB}$/$\overline{PACK}$	t_{H_PCS}	—	0	—	—	ns
PIOA Setup Time for $\overline{PSTB}$/$\overline{PACK}$	t_{S_PIOA}	—	0	—	—	ns
PIOA Hold Time for $\overline{PSTB}$/$\overline{PACK}$	t_{H_PIOA}	—	3	—	—	ns
Required $\overline{PACK}$ Time	t_{W_PACK}	—	30 + t_{CYC}	—	—	ns
Required $\overline{PSTB}$ Time	t_{W_PSTB}	—	30 + 2 t_{CYC}	—	—	ns
PD Setup Time for $\overline{PSTB}$	t_{S_PD}	—	–t_{CYC}	—	—	ns
PD Hold Time for $\overline{PSTB}$	t_{H_PD}	—	8	—	—	ns

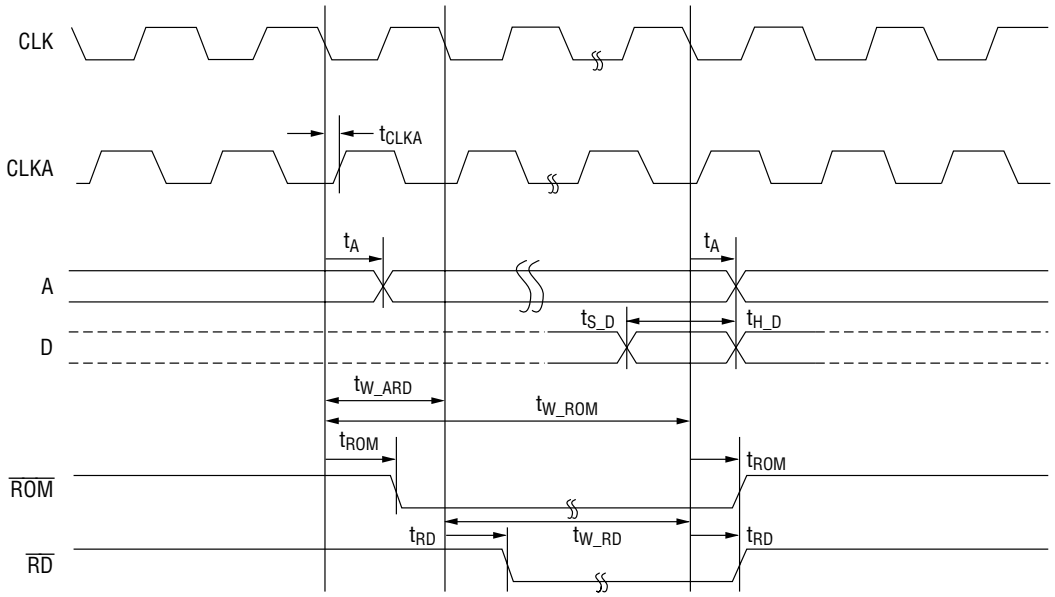
Note 1 According to DRAM configuration
Note 2 By the DRAM access timing
Note 3 In the case of writing, increased by 1 clock when X bit = 0 and by 2 clocks when X bit = 1.
Note 4 Flash memory access timing is the same with the SRAM timing.

TIMING DIAGRAM

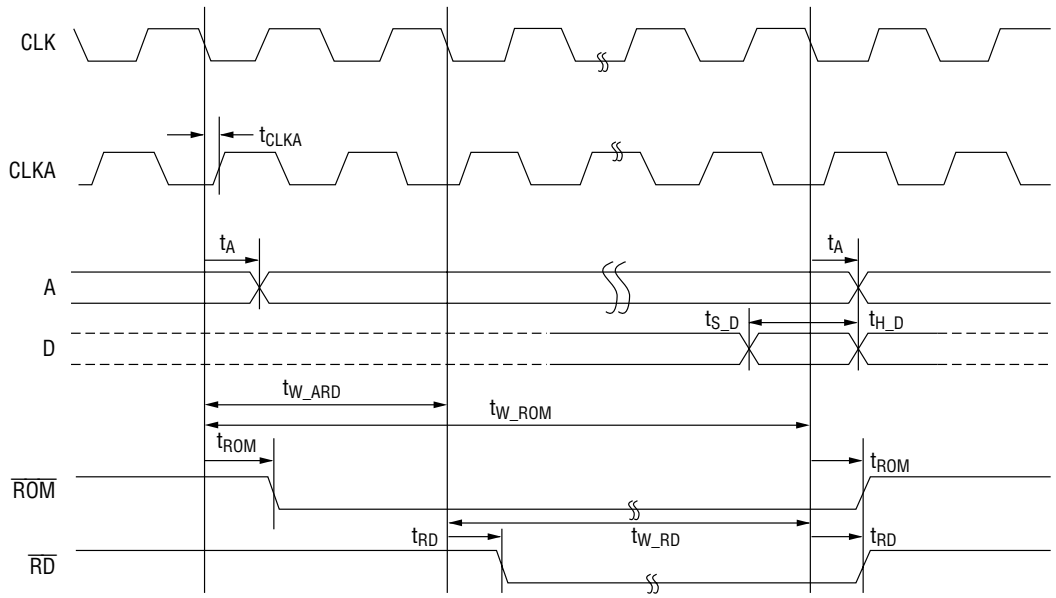
Clock And Reset



ROM Read

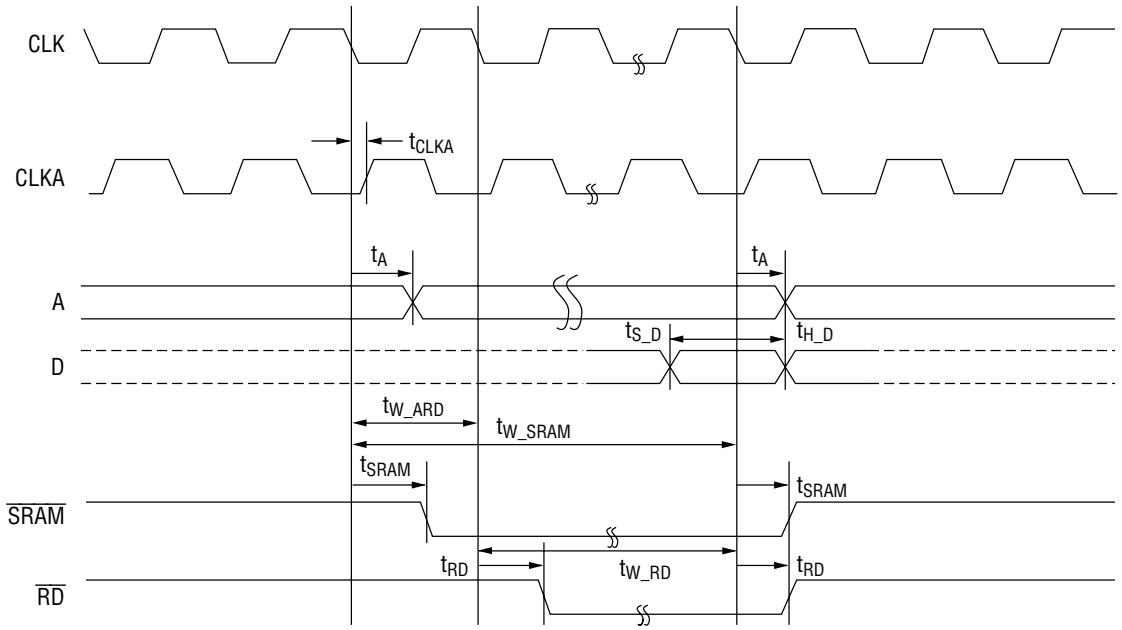


3τ/4τ Access

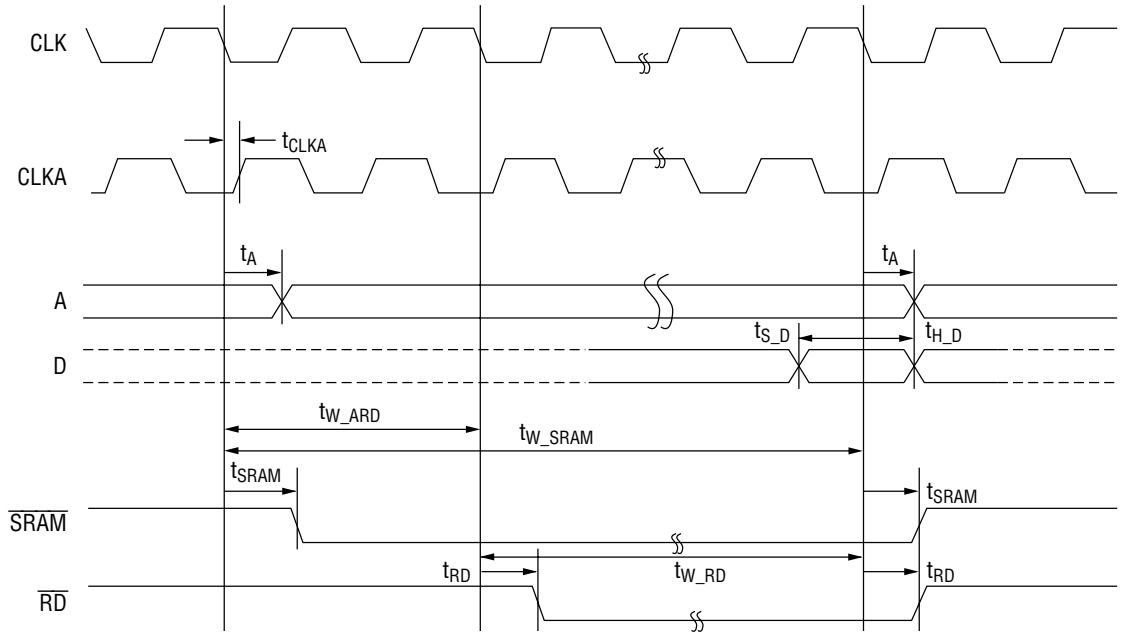


5τ/6τ/8τ/10τ/12τ Access

SRAM Read

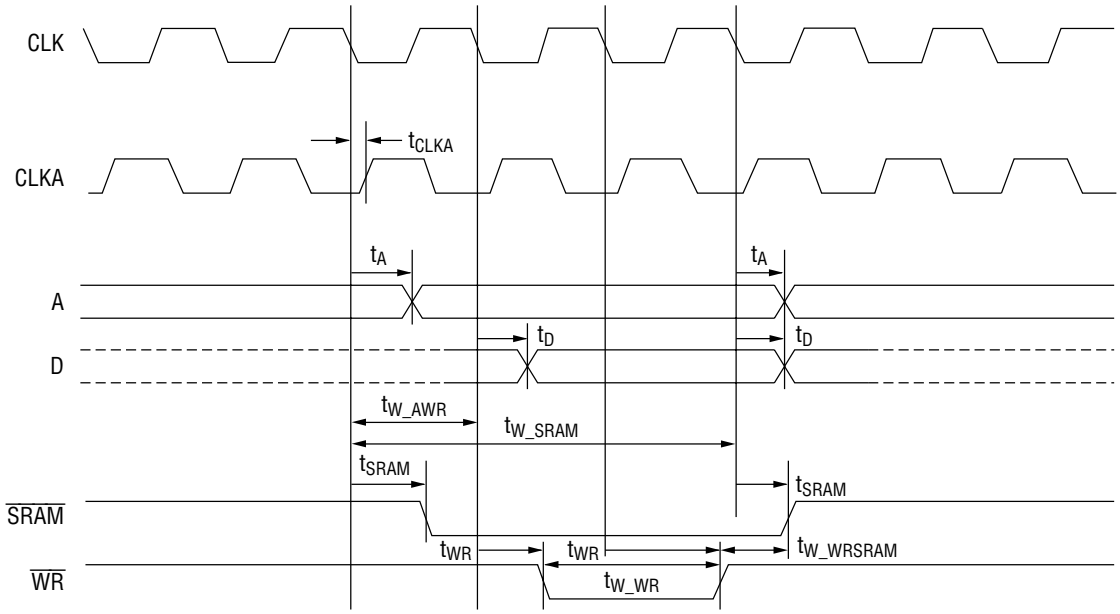


3τ/4τ Access

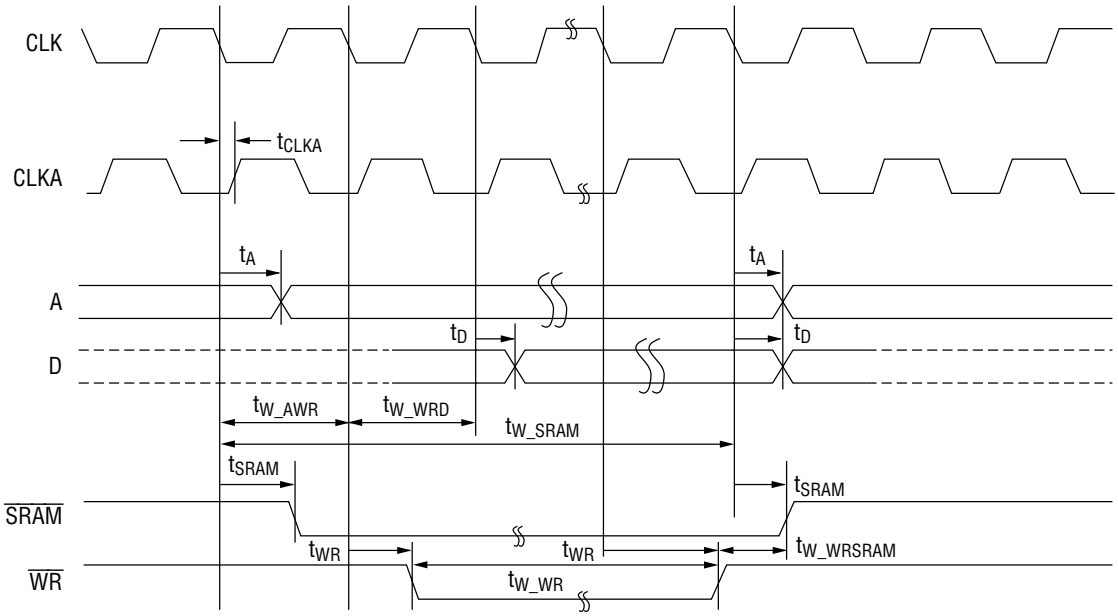


5τ/6τ/8τ/10τ/12τ Access

SRAM Write

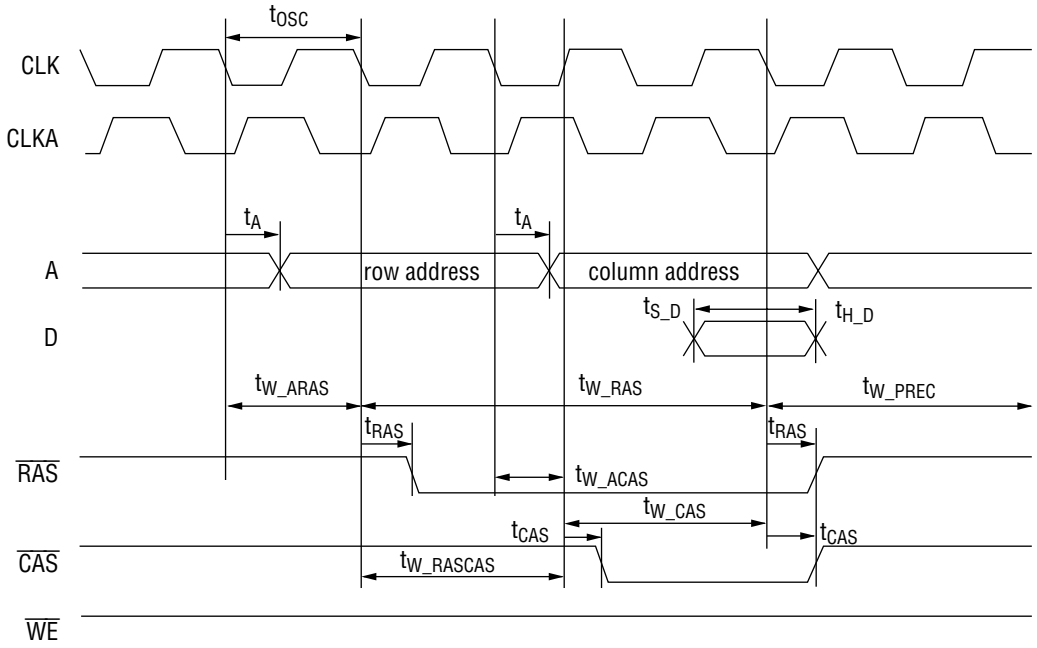


3τ Access

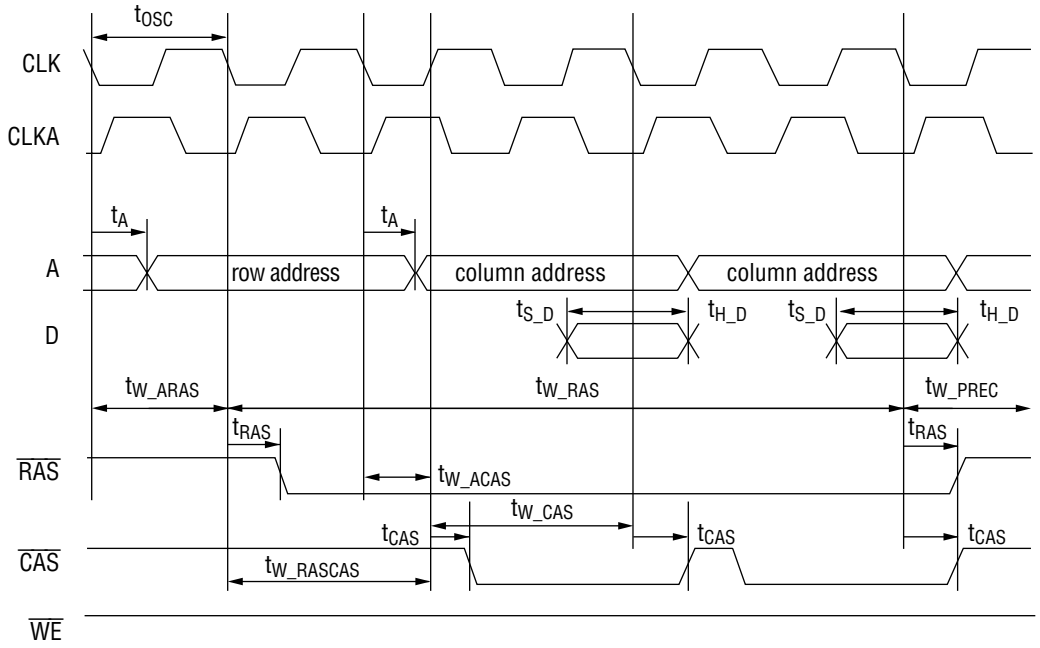


4τ/5τ/6τ/8τ/10τ/12τ Access

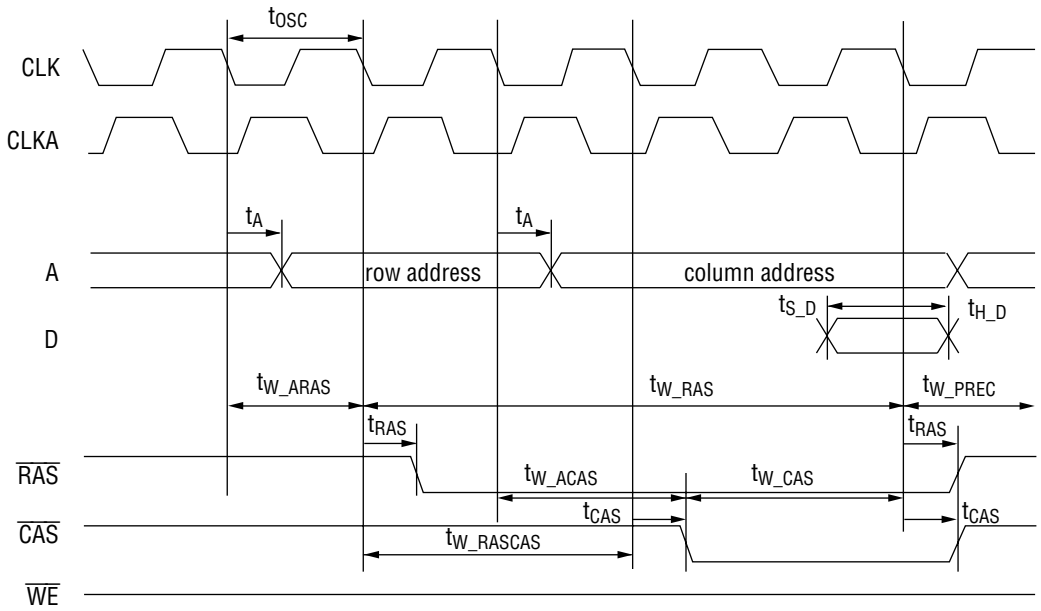
DRAM Read



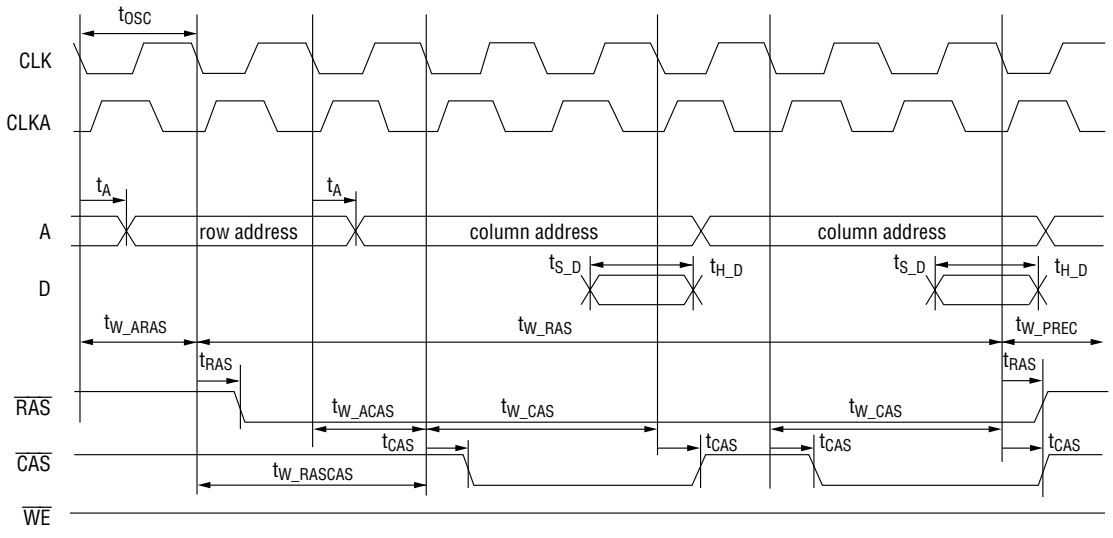
2nτ Access (Fast Page Mode)



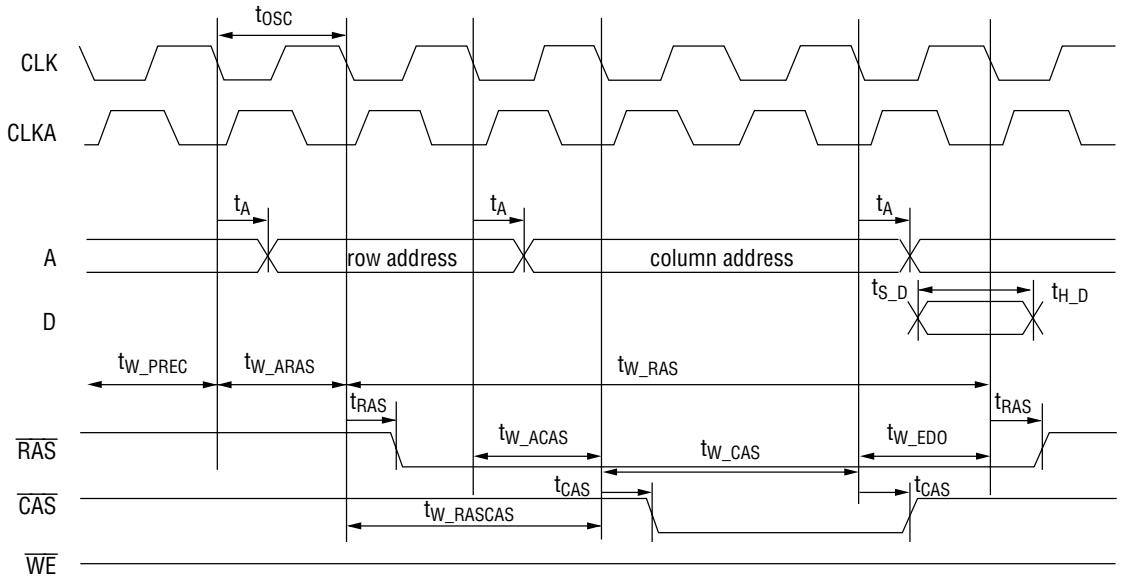
2nτ Access (Fast Page Mode)



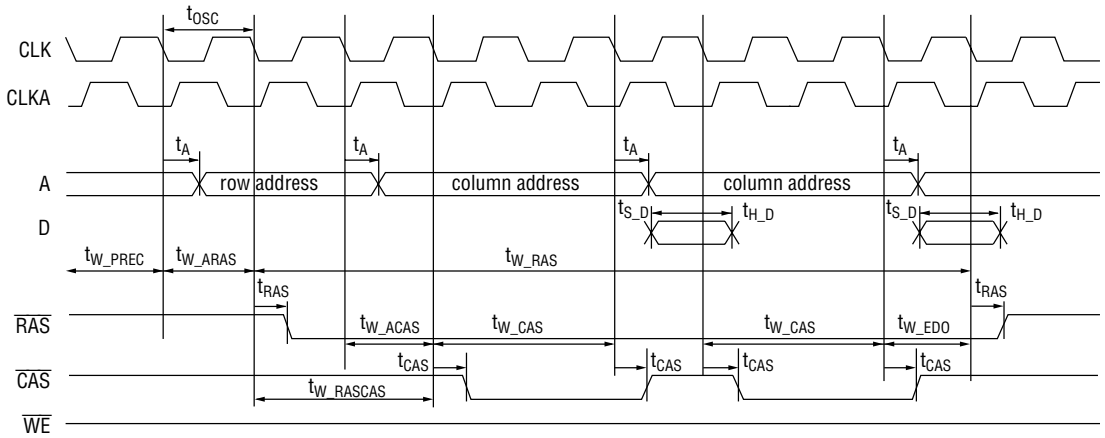
3nτ Access (Fast Page Mode)



3nτ Access (Fast Page Mode)

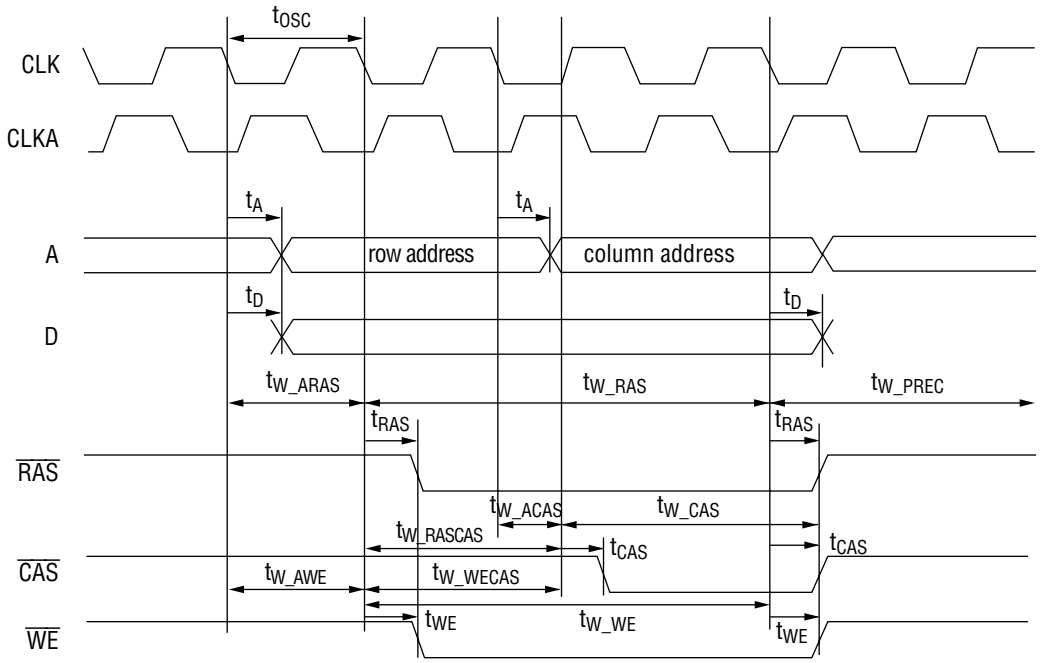


3nτ Access (Hyperpage Mode)

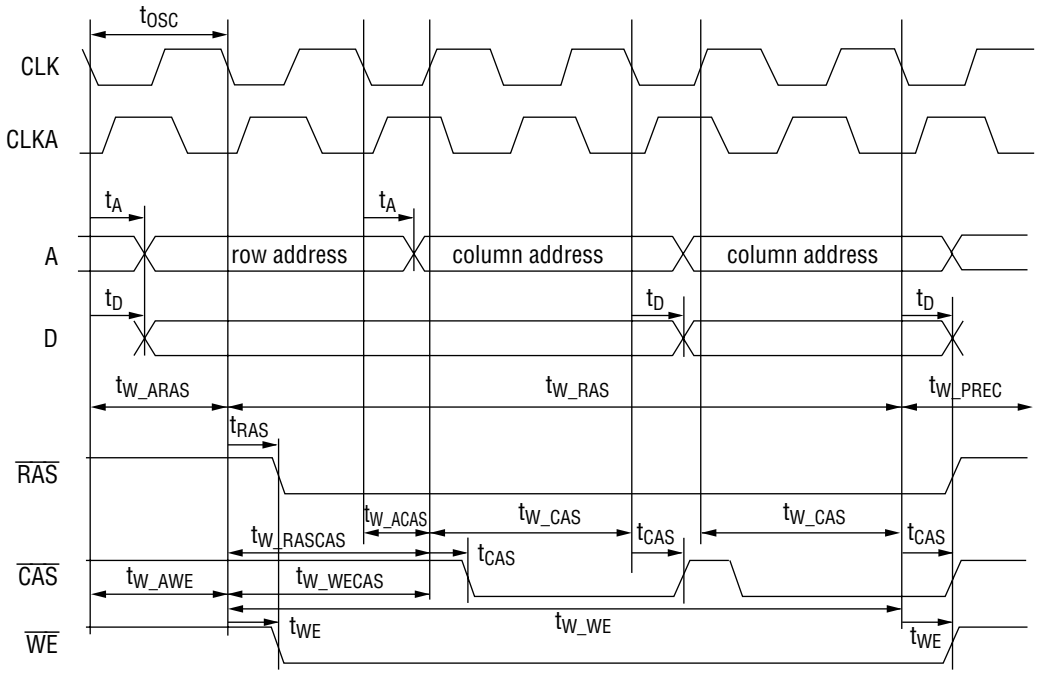


3nτ Access (Hyperpage Mode)

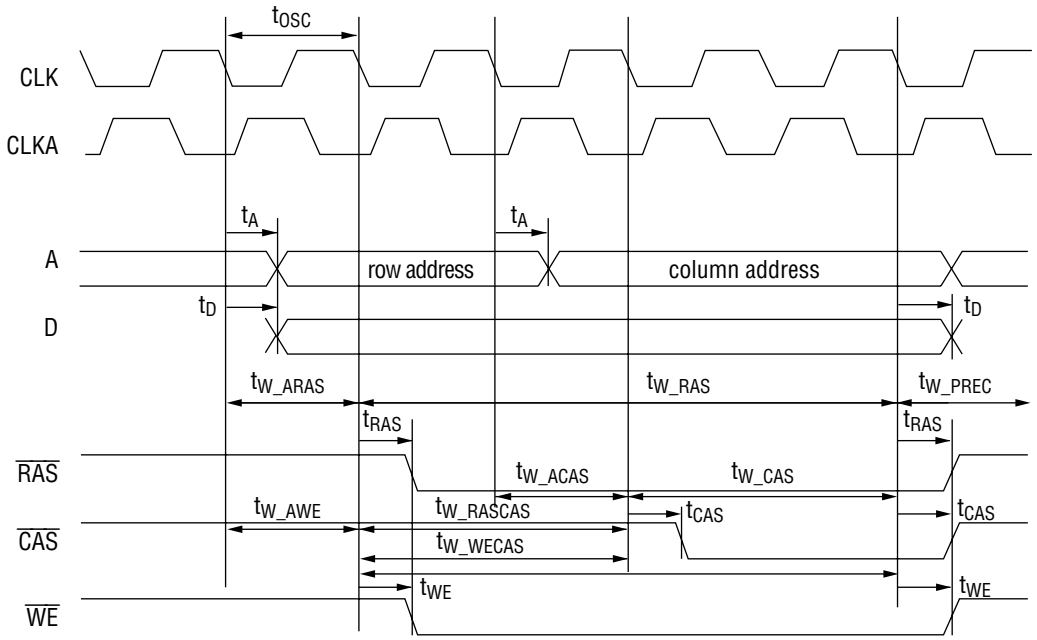
DRAM Write



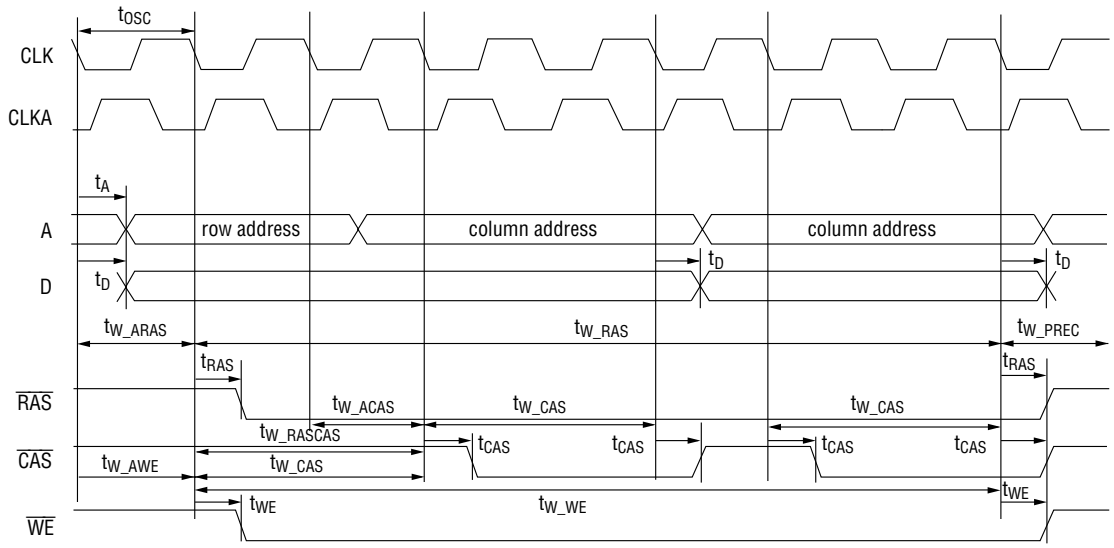
2nτ Access (Fast Page Mode)



2nτ Access (Fast Page Mode)

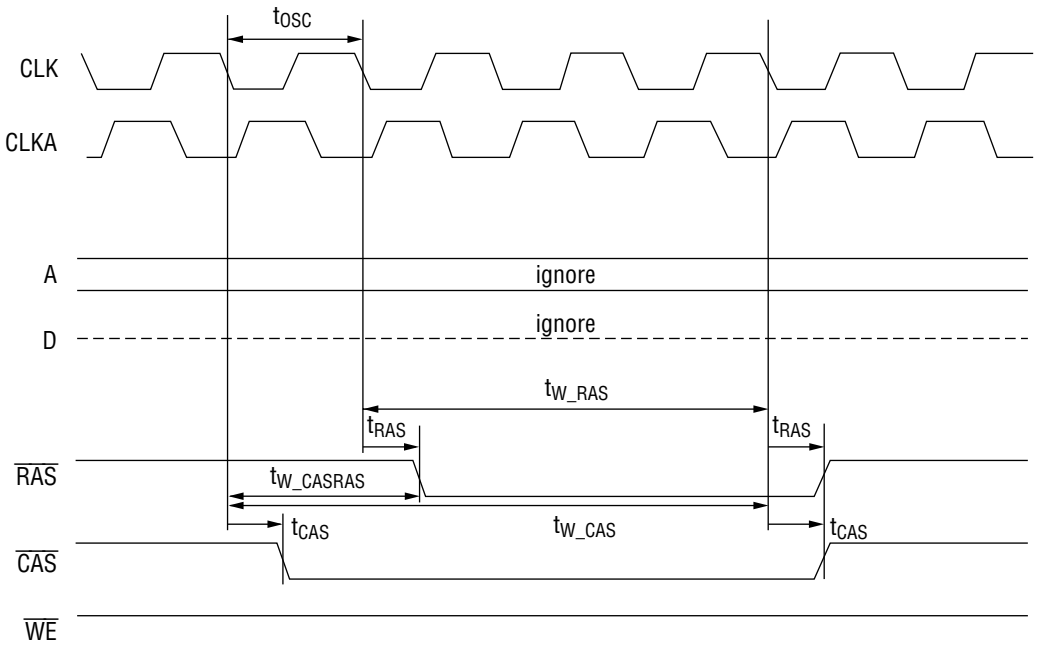


3nτ Access (Fast Page Mode/Hyperpage Mode)

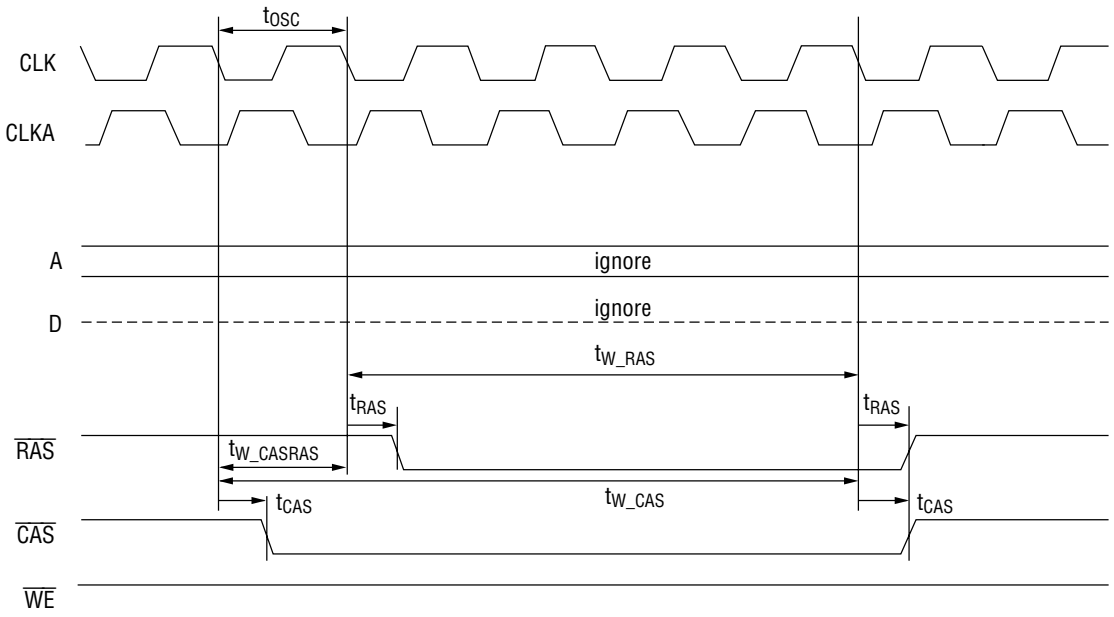


3nτ Access (Fast Page Mode/Hyperpage Mode)

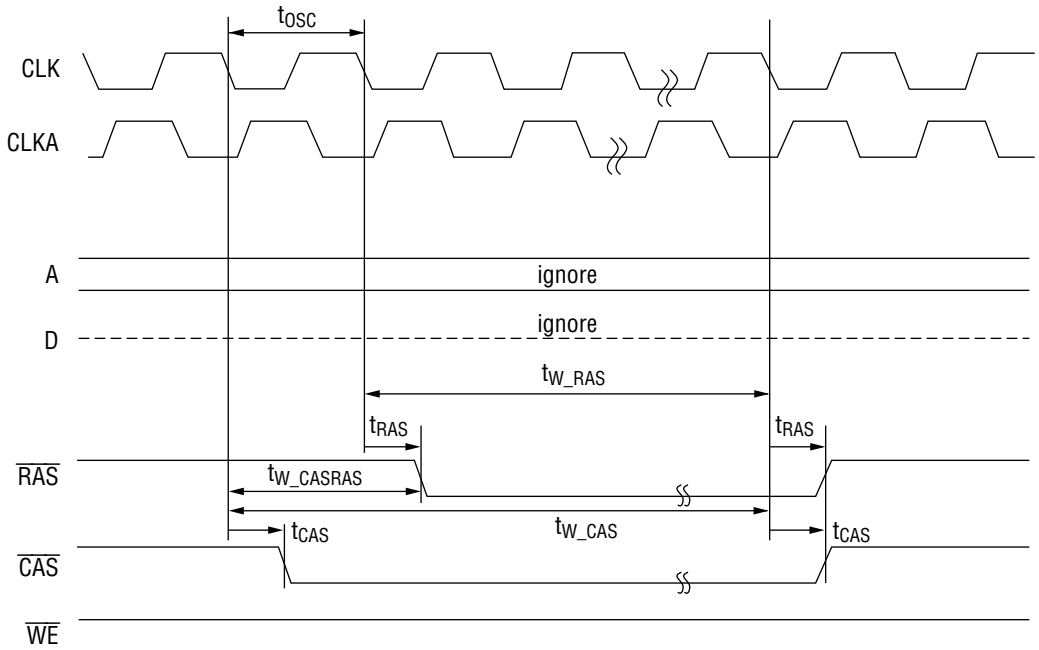
DRAM Refresh



2nτ CAS-Before-RAS Refresh

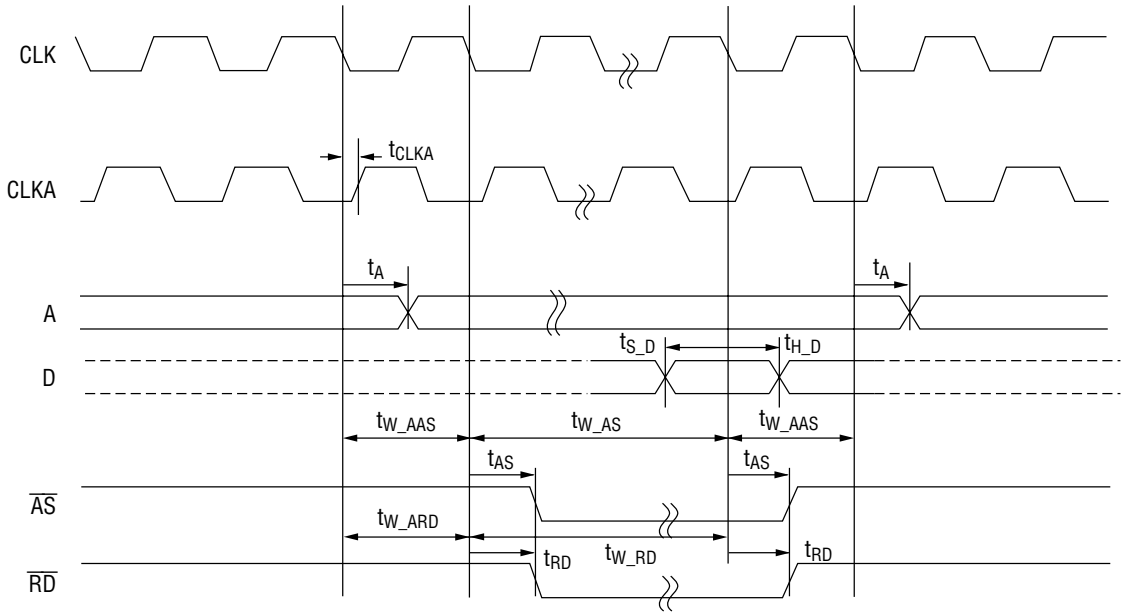


3nτ CAS-Before-RAS Refresh

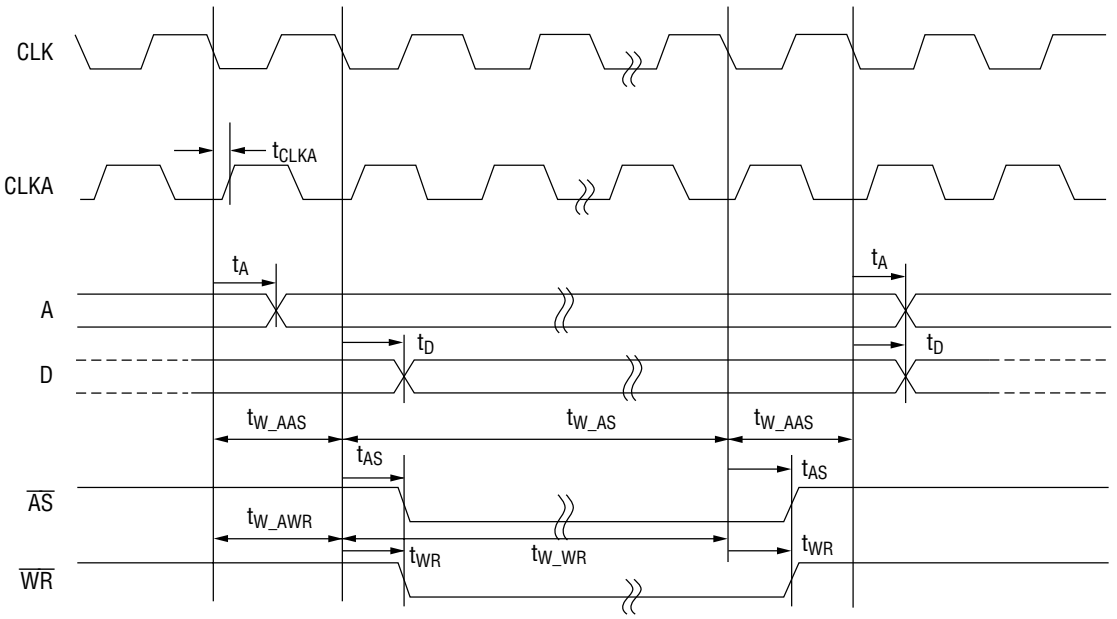


CAS-Before-RAS Self-Refresh

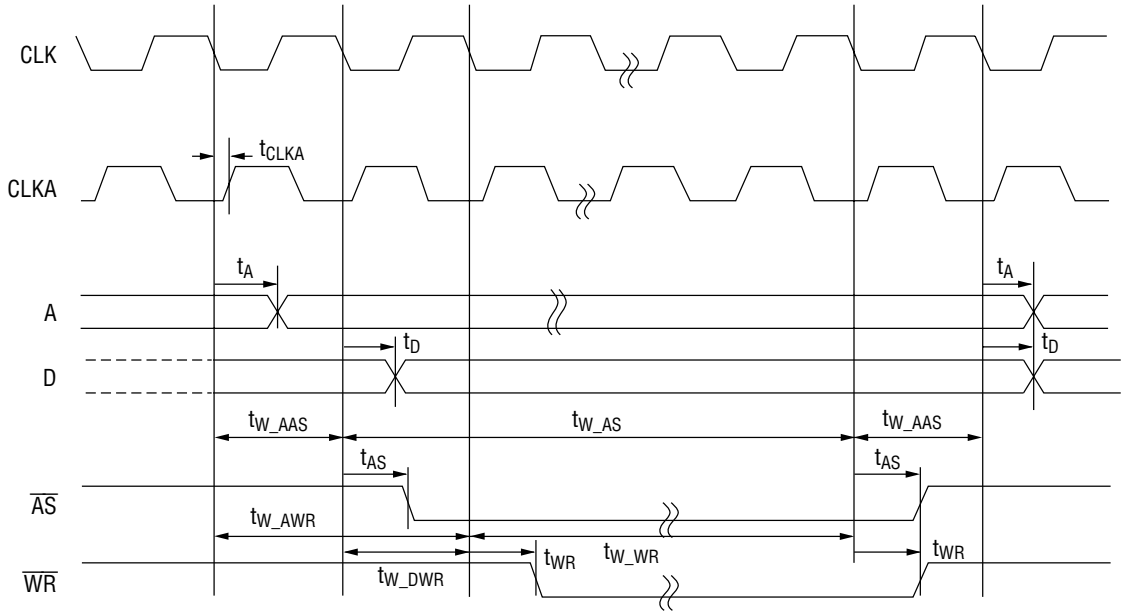
General Device Access



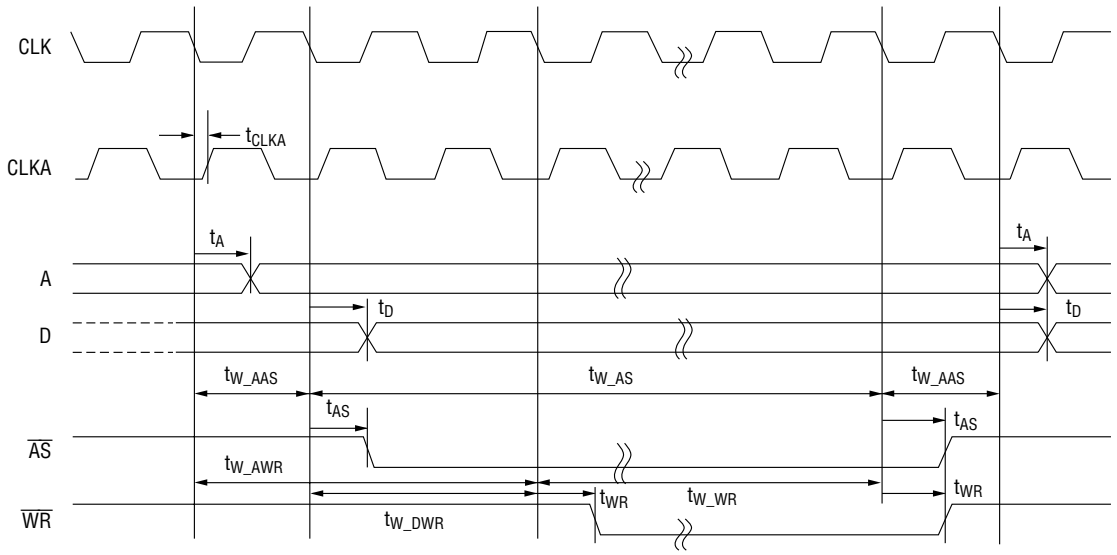
Bus Read



Bus Write (When DS bit in the SCR register is "0")

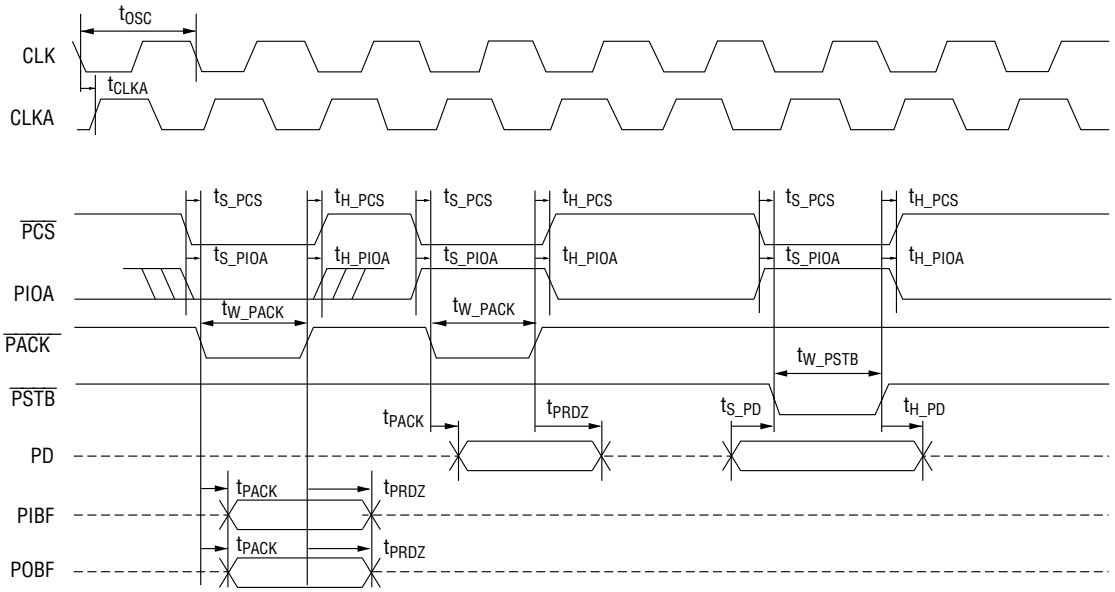


Bus Write (When DS bit is "1" and X bit is "0" in the SCR register)

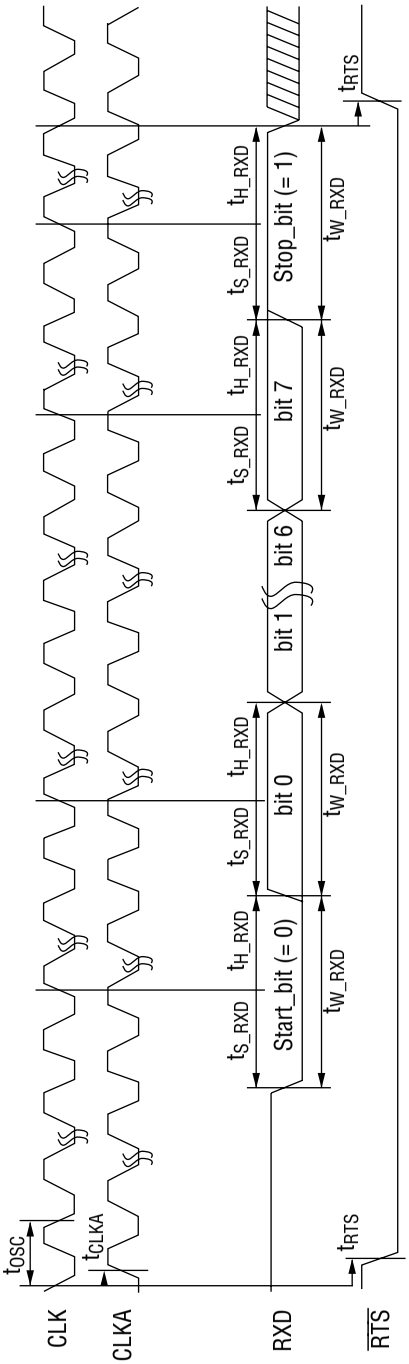


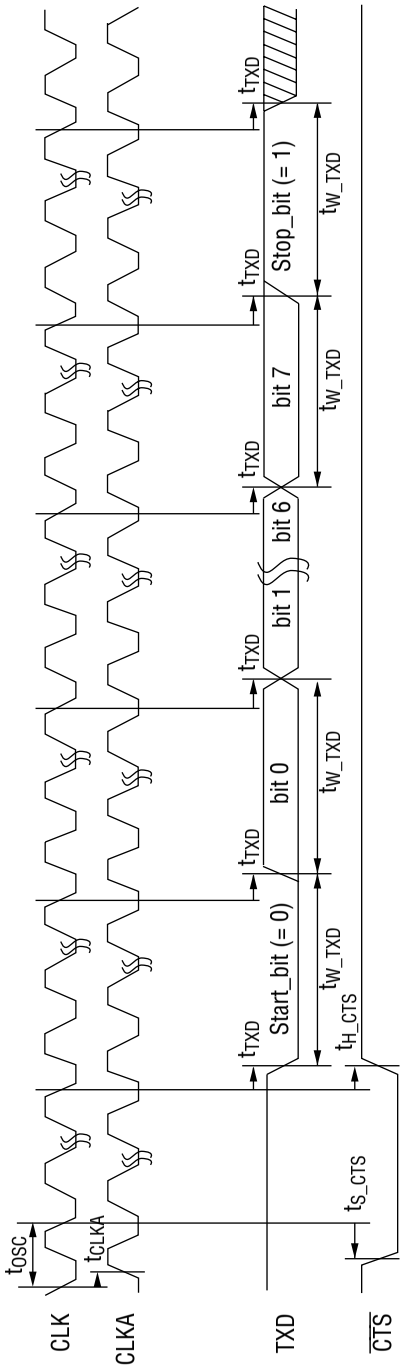
Bus Write (When DS bit is "1" and X bit is "1" in the SCR register)

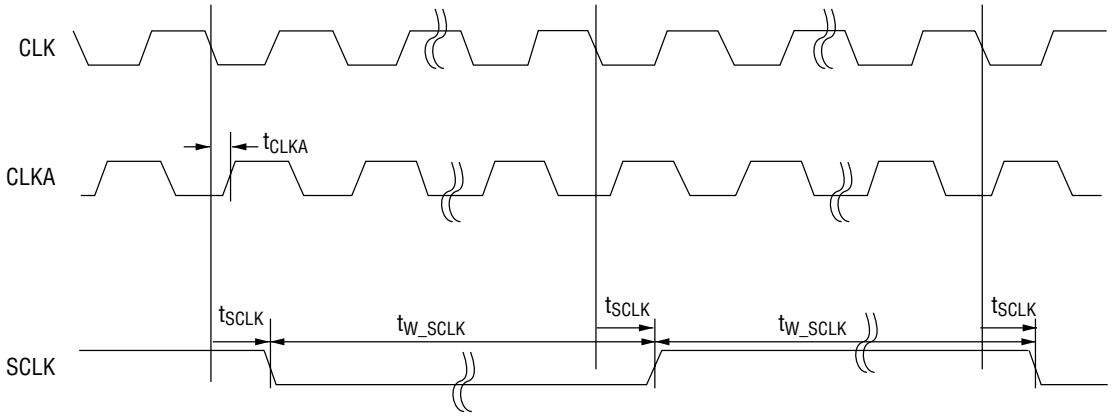
Parallel Interface



Serial Interface

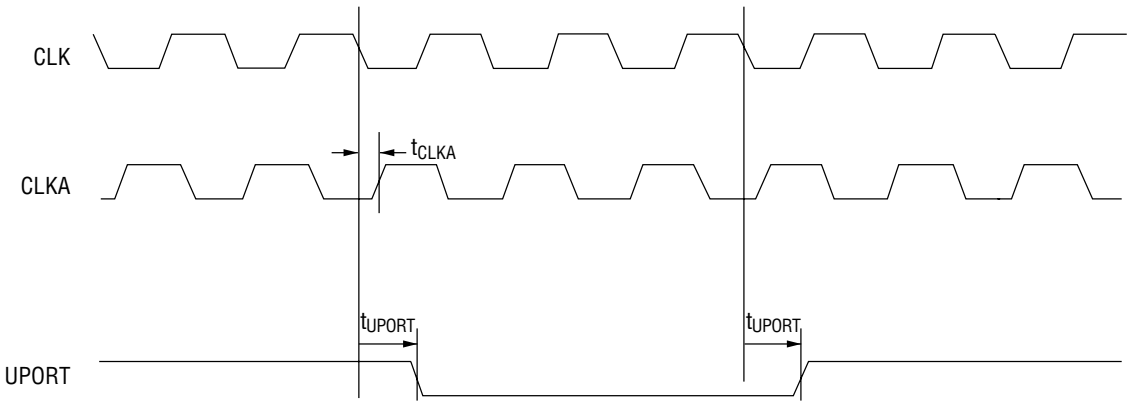






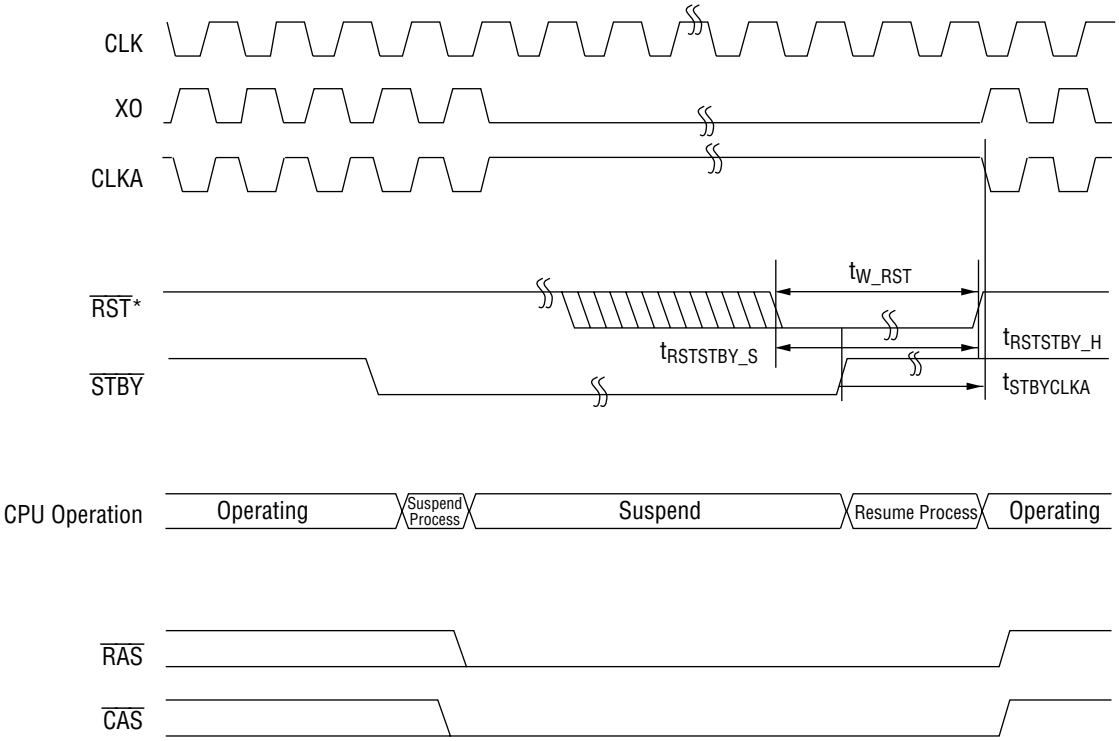
Synchronous Transfer Output

General Port Output



General Port Output

Standby Operation

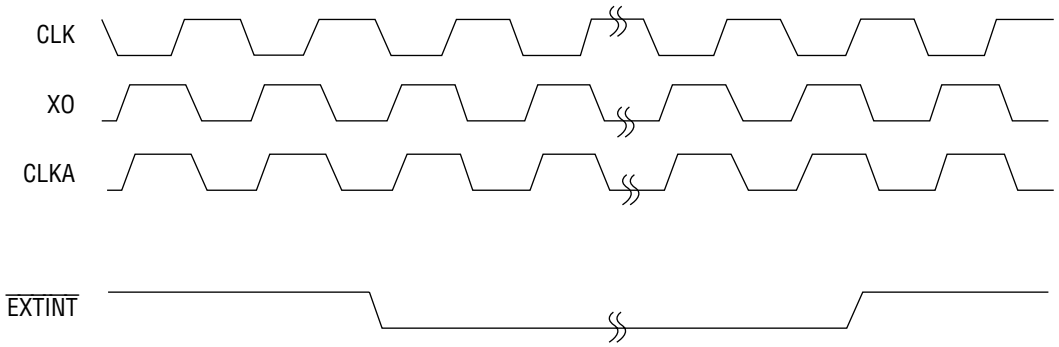


Maintain the pin level on the $\overline{STBY}$ signal until the CPU has completed its suspend process and clock signal CLKA has stopped.

After the $\overline{STBY}$ signal is released, the CPU will not resume until oscillation has stabilized (1024 t_{CYC}).

* The $\overline{RST}$ signal is not necessary for self-refresh DRAM.

Interrupt Process



The external interrupt signal $\overline{\text{EXTINT}}$ requests an interrupt to the CPU. The pin level on $\overline{\text{EXTINT}}$ must be maintained until the CPU accepts the interrupt. Also, be sure to clear the interrupt source within the interrupt routine.

FUNCTIONAL DESCRIPTION

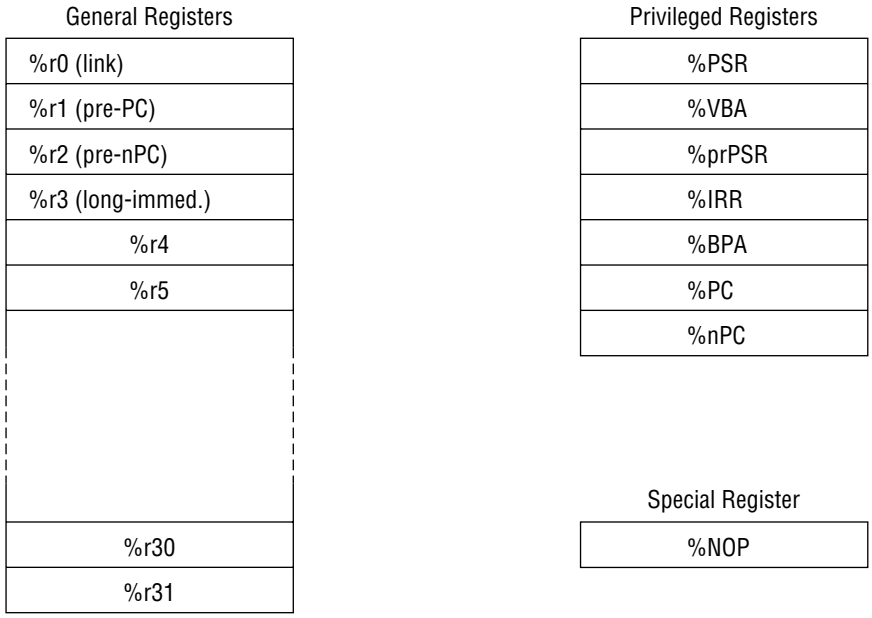
CPU Core

1. Features

The SCP (Speech Control Processor) uses a CPU core with an Oki-original 32-bit RISC architecture.

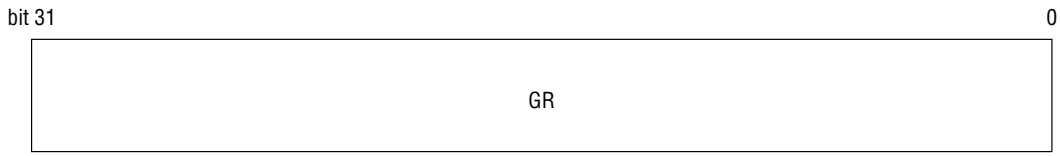
2. Register Configuration

The CPU core registers are configured as 32 words for general registers, 7 words for privileged registers, and 1 word for a special register.



2.1 General Registers

The general registers are a set of 32 registers with 32-bit width. Of these registers %r0 to %r3 can be used as general registers, but they do have special functions pre-assigned by the system. Registers %r4 to %r31 can be used freely. Contents are undefined after reset.



%r0: Link register (stores subroutine return address). Also stores %PC+4 during bl instruction execution.
%r1: Stores value of %PC when an exception, interrupt, or trap is accepted.

%r2: Stores value of %nPC when an exception, interrupt, or trap is accepted.

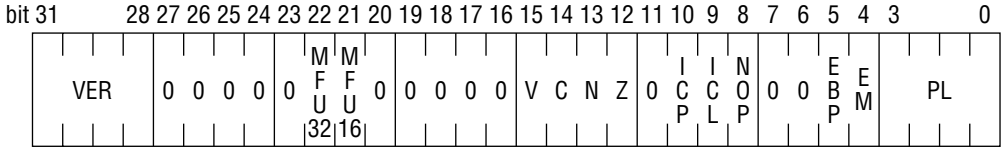
%r3: Stores the immediate value of SETLI (Set Long Immediate) instructions.

2.2 Privileged Registers

Reads are allowed at any processor level (Processor Level: 0 = user mode, 1 or above = supervisor mode), but write accesses are allowed only when the processor level is supervisor mode. The privileged registers are configured as 7 words, and are used primarily for processor control. If the processor attempts a write access to a privileged register while in user mode, then the instruction will not be executed and a privileged instruction exception will be issued.

2.2.1 PSR (Processor Status Register)

This register sets and displays the state of the processor.



• bit[31:28] VER: Version (read-only)

Indicates the CPU core version. Currently fixed to "3".

• bit[22] MFU32 (read-only)

Indicates whether the 32-bit multiplier unit is present ("1") or not ("0"). This is "0" for the MSM7630.

• bit[21] MFU16 (read-only)

Indicates whether the 16-bit multiplier unit is present ("1") or not ("0"). This is "1" for the MSM7630.

• bit[15] V: Overflow (read-only)

Indicates that execution of an addition or subtraction instruction resulted in an arithmetic overflow.

• bit[14] C: Carry (read-only)

Indicates that execution of an addition or subtraction instruction resulted in an arithmetic carry or borrow.

• bit[13] N: Negative (read-only)

Indicates that execution of an addition or subtraction instruction resulted in a negative value (bit[31] is "1").

- bit[12] Z: Zero (read-only)

  Indicates that execution of an addition or subtraction instruction resulted in a zero value (bit[31:0] are all "0").

- bit[10] ICP: Instruction Cache Purge (read/write)

  Invalidates all instruction cache entries. Writing "1" to this bit purges the contents of the instruction cache. After this process (after one cycle) this bit is automatically cleared to "0" by hardware. The instruction cache is purged during reset.

- bit[9] ICL: Instruction Cache Lock (read/write)

  Freezes all instruction cache entries. After "1" is written to this bit, instruction cache contents are frozen and then instruction execution continues. This bit will be "1" after reset.

- bit[8] NOP: Non-Operation (read-only)

  When set to "1", forces the next instruction to a NOP regardless of the instruction. There is no way to directly set this bit to "1". This bit will be "0" after reset.

- bit [5] EBP: Breakpoint Trap Enable (read/write)

  Enables breaks. If this bit is set to "1", then a trap will occur when the value of the instruction execution address (%PC) equals the value of the breakpoint address (%BPA). The instruction that generated the break will not be executed. This bit will be "0" after reset.

- bit[4] EM: Master Enable (read/write)

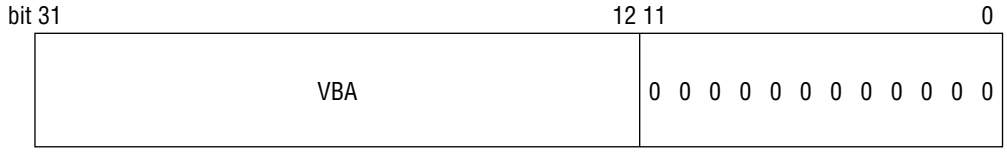
  Disables all exceptions, interrupts, and traps. This bit automatically becomes "0" at the point when the processor accepts an exception, interrupt, or trap. While this bit is "0", further exceptions, interrupts or traps will not be accepted, with instruction execution continuing in the normal instruction sequence. An instruction must be used to return this bit to "1". It will be "0" after reset.

- bit[3:0] PL: Processor Level (read/write)

  Sets and provides the processor's instruction execution level. Processor levels are 0-15. An external interrupt will be accepted if its level has a higher priority than the processor level at that time. External interrupt levels are 1-16, so when PL is 0 all external interrupts will be accepted, and when PL is 1 external interrupts of level 2 and above will be accepted. When an external interrupt is accepted, the processor level will become the same as the external interrupt level. For example, if PL is 5 and a level 7 external interrupt is accepted, then PL will transition to 7 at that point. When PL is restored to its previous state, its saved value in %prPSR will be restored to %PSR. Alternatively PL can be set to its previous value explicitly by an instruction in the interrupt process routine. However, %PSR is a privileged register, so writes are only permitted in supervisor mode. PL will be set to 15 after reset.

2.2.2 VBA: Vector Base Address (read/write)

This read/write register sets the leading address of the dispatch table (vector table) to exception, interrupt, and trap process routines.



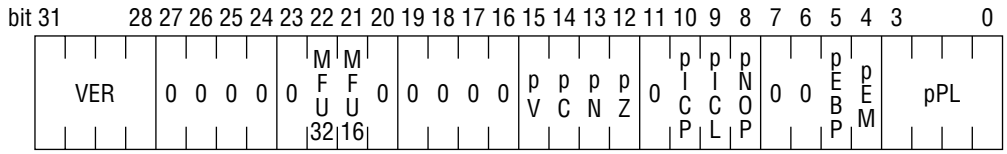
The dispatch table is 256 entries of 4K bytes size, with 16 bytes (4 instructions) save for each entry's dispatch routine. Entry points are generated by an OR operation with this register, so they are set at 4K-byte boundaries. As a result, only the upper 20 bits of an argument will be written to the VBA register (the lower 12 bits will be ignored).

Entry_point = VBA[31:12] || (vector_number << 4)

This register is undefined after reset.

2.2.3 prPSR: Pre-Processor Status Register (read/write)

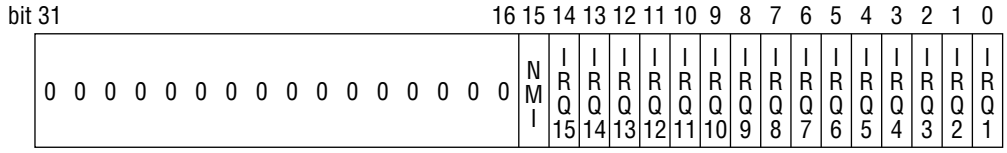
This read/write register saves the value of %PSR at the time an exception, interrupt, or trap is accepted. In order to accept overlapping exceptions, interrupts, and traps, the value of %prPSR must be pushed on a stack and then EM of %PSR must be set to "1".



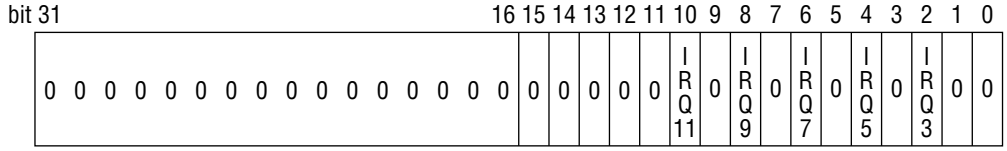
The upper 16 bits of %prPSR are always identical to %PSR. Refer to the descriptions of the same bit positions in %PSR for an explanation of %prPSR bits.

2.2.4 IRR: Interrupt Request Register (read-only)

This register indicates whether there is an interrupt request at each of the 16 levels of external interrupts. It is read-only, and shows interrupt requests regardless of PL (processor level). The IRR value will continue until an interrupt source is released.

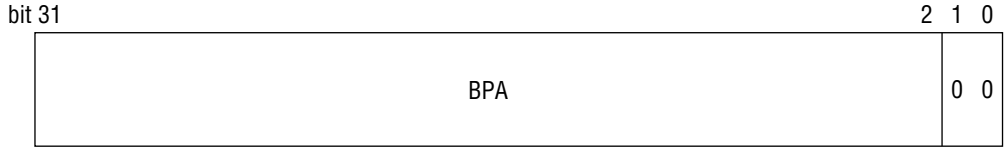


The MSM7630 uses only 6 interrupts of the 16 interrupt levels.



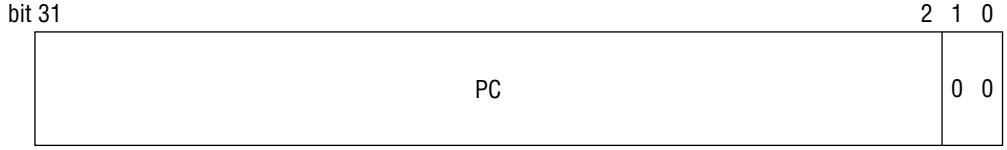
2.2.5 BPA: Breakpoint Address (read/write)

This read/write register sets and shows the instruction address (byte address) where a breakpoint trap occurred. The lowest 2 bits will always be "0". When EBP of %PSR is "1", a trap will be generated immediately before execution of the instruction at the breakpoint set by this register. This register will be undefined after reset.



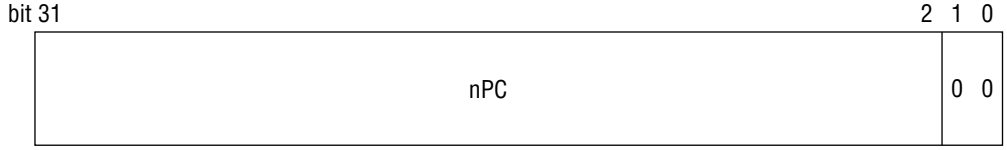
2.2.6 PC: Program Counter (read-only)

This read-only register provides the instruction address (byte address) in the execution phase. Its lowest 2 bits will always be "0".



2.2.7 nPC: Next Program Counter (read-only)

This read-only register provides the instruction address (byte address) in the instruction decode phase. Its lowest 2 bits will always be "0".

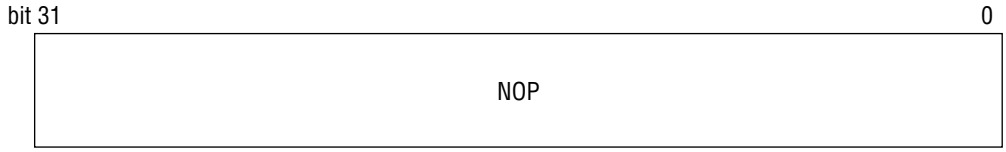


2.3 Special Registers

These are not privileged registers, but they are special registers used for specific functions.

2.3.1 NOP: Non-Operation (read/write)

When this register is specified as a destination register, execution results will not be stored anywhere. When specified as a source register, it will read as an undefined value.

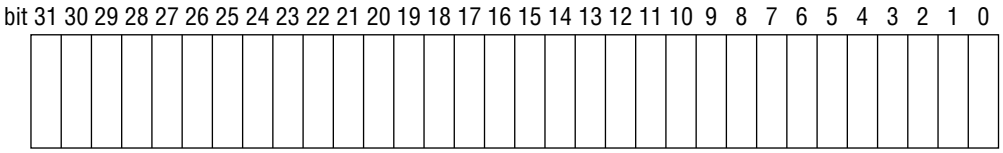


3. Data Formats

There are two data format types: one for internal processor core calculations and one for memory accesses.

3.1 Internal Data Format

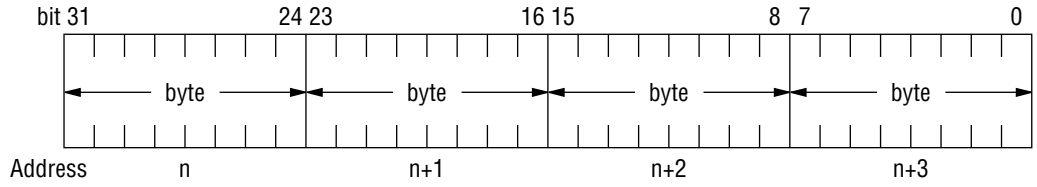
The CPU core handles all data as 32 bits (word format). Therefore, when the format of data stored in memory is byte (8 bits) or half-word (16 bits) it must be used internally as 32-bit data through a signed load instruction or unsigned load instruction. Similarly when internal core processing results are stored to memory, a store instruction corresponding to the data format in memory must be executed. Also, bit addresses specified for bit test instructions and bit manipulation instructions are shown in the diagram below.



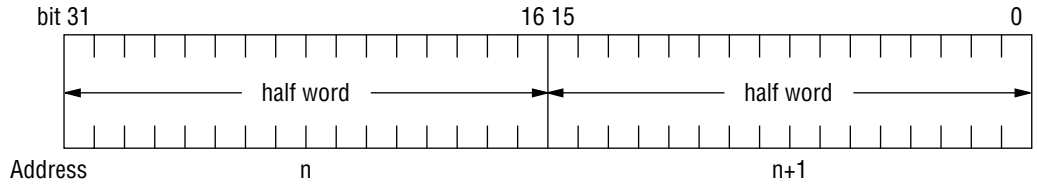
3.2 Memory Data Format

The following memory data formats are supported: byte (8 bits), half-word (16 bits), and word (32 bits). Memory addresses are always byte addresses regardless of data format. However, half-word accesses must be on 16-bit boundaries (least significant bit is "0"), and word accesses must be on 32-bit boundaries (least significant 2 bits are "00"). If a load or store instruction execution attempts a memory access that violates these boundaries, then a data address invalid exception will occur. Memory addressing is big-endian. The diagrams below show memory data formats for byte data access, half-word data access, and word data access.

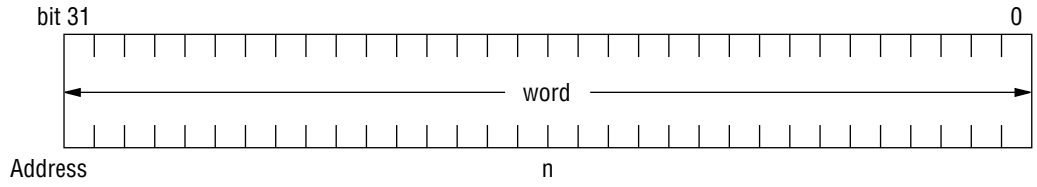
Byte Data Access



Half-Word Data Access



Word Data Access



3.3 Memory Addressing Modes

Memory addresses are byte addresses, so memory addressing is performed with three types of load instructions and two types of store instructions. Swap instructions have the same memory addressing as store instructions.

3.3.1 Load Instruction Addressing

1. Base + Index

The effective address (EA) is obtained by adding the values of any two general registers %r0-31 specified.

EA = [reg_S1 + reg_S2]

2. Base + Displacement

The effective address (EA) is obtained by adding the value of any general register %r0-31 specified and a displacement given by the instruction's immediate value field.

EA = [reg_S1] + offS

3.3.2 Store Instruction Addressing

1. Base + Displacement

The effective address (EA) is obtained by adding the value of any general register %r0-31 specified and a displacement given by the instruction's immediate value field.

EA = [reg_S1] + offS

4. Instruction Set

All instructions are fixed 32-bit length.

Category	Instruction	Function
Unconditional branch	b{,x}?	Unconditional branch
	bl{,x}?	Unconditional branch to subroutine
Conditional branch	bt{,x/t} S1,?	Conditional Branch
	bf{,x/t} S1,?	Conditional Branch
	jlr {,x} S2,D*	Conditional branch to subroutine
	jlrt {,x/t} S1,S2,D*	Conditional branch to subroutine
	jlrf {,x/t} S1,S2,D*	Conditional branch to subroutine
	rt S2	Return from subroutine
Bit test	btst1 S1,S2/immU,D	Bit test
	btst0 S1,S2/immU,D	Bit test
Comparison	cmpeq S1,S2/immS,D	Comparison [=]
	cmple S1,S2/immS,D	Comparison [signed: ≤]
	cmplt S1,S2/immS,D	Comparison [signed: <]
	cmpls S1,S2/immS,D	Comparison [unsigned: ≤]
	cmpc S1,S2/immS,D	Comparison [unsigned: <]
	cmpne S1,S2/immS,D	Comparison [≠]
	cmpgt S1,S2/immS,D	Comparison [signed: >]
	cmpge S1,S2/immS,D	Comparison [signed: ≥]
	cmphi S1,S2/immS,D	Comparison [unsigned: >]
	cmpnc S1,S2/immS,D	Comparison [unsigned: ≥]
Trap	trap vct	Transfer to trap toutine
Arithmetic/logical operation	add S1,S2/immS9,D	Add
	sub S1,S2/immS9,D	Subtract
	adc S1,S2/immS9,D	Add with carry
	sbc S1,S2/immS9,D	Subtract with carry
	and S1,S2/immS9,D	Logical AND
	or S1,S2/immS9,D	Logical OR
	xor S1,S2/immS9,D	Exclusive OR
	sbr S1,S2/immS12,D	Subtract
Extend	extu S1,S2/immU,D	MSB extend
	ext S1,S2/immU,D	MSB extend
Shift	sl S1,S2/immS,D	Logical shift
	rot S1,S2/immS,D	Logical rotate
	slr S1,S2/immS,D	Logical shift
	sar S1,S2/immS,D	Arithmetic shift
Bit manipulation	brst S1,S2/immU,D	Set bit to "0"
	bset S1,S2/immU,D	Set bit to "1"
	bnot S1,S2/immU,D	Invert bit
	brst %psr,4/5,%psr	Set bit to "0"
	brst %psr,4/5,%psr	Set bit to "1"

Category	Instruction	Function
Register-register move	mov S,D	Move
	movh S1,D'	Move upper bits
Store immediate value	seti imm17S,D	Store immediate value
	setih const16,D'	Store immediate value to upper 16 bits
	setli const25	Store immediate value left-shifted 7 bits to %r3
Store	sb S2/immS,[S1+offS]	Byte store
	shw S2/immS,[S1+offS]	Half-word store
	sw S2/immS,[S1+offS]	Word store
Swap	swap S2,[S1+offS]	Swap
Load	lb [S1+offS],D'	Byte load
	lhw [S1+offS],D'	Half-word load
	lw [S1+offS],D'	Word load
Multiplication	mul0 S1,S2/immS,D'	Signed multiply
	mul16 S1,S2/immS,D'	Signed multiply
	mul32 S1,S2/immS,D'	Signed multiply
	mulu0 S1,S2/immS,D'	Unsigned multiply
	mulu16 S1,S2/immS,D'	Unsigned multiply
	mulu32 S1,S2/immS,D'	Unsigned multiply

Multiply instructions need two clocks for execution time.
The MSM7630 can only use the mul0 and mulu0 instructions of the multiplication instructions.

5. Exceptions, Traps, and Interrupts

The CPU core of SCP provides error exceptions, traps, external interrupts, and software traps (by trap instructions). Each type has a corresponding interrupt priority level and instruction dispatch address.

Source	Vector Number	Branch Address	Priority	Synchronous/Asynchronous (Sense)
System reset		0x00000000	0	Asynchronous (level)
CPU reset (INIT)	0	VBA+0x000	1	Asynchronous (edge)
Instruction access exception	1	VBA+0x010	2	Synchronous
Instruction address invalid exception	2	VBA+0x020	3	Synchronous
Reserved instruction exception	3	VBA+0x030	4	Synchronous
Privileged instruction exception	4	VBA+0x040	5	Synchronous
Data address invalid exception	5	VBA+0x050	8	Asynchronous (edge)
Data access exception	6	VBA+0x060	9	Asynchronous (edge)
Reserved	7	VBA+0x070		
Breakpoint trap	8	VBA+0x080	6	Synchronous
Reserved	9 to 32	VBA+0x090 to VBA+0x200		
External interrupt 1	33	VBA+0x210	25	Asynchronous (level)
External interrupt 2	34	VBA+0x220	24	Asynchronous (level)
External interrupt 3	35	VBA+0x230	23	Asynchronous (level)
External interrupt 4	36	VBA+0x240	22	Asynchronous (level)
External interrupt 5	37	VBA+0x250	21	Asynchronous (level)
External interrupt 6	38	VBA+0x260	20	Asynchronous (level)
External interrupt 7	39	VBA+0x270	19	Asynchronous (level)
External interrupt 8	40	VBA+0x280	18	Asynchronous (level)
External interrupt 9	41	VBA+0x290	17	Asynchronous (level)
External interrupt 10	42	VBA+0x2a0	16	Asynchronous (level)
External interrupt 11	43	VBA+0x2b0	15	Asynchronous (level)
External interrupt 12	44	VBA+0x2c0	14	Asynchronous (level)
External interrupt 13	45	VBA+0x2d0	13	Asynchronous (level)
External interrupt 14	46	VBA+0x2e0	12	Asynchronous (level)
External interrupt 15	47	VBA+0x2f0	11	Asynchronous (level)
External interrupt 16 (NMI)	48	VBA+0x300	10	Asynchronous (edge)
TRAP instruction	0 to 255	VBA+0x000 to VBA+0xff0	7	Synchronous

The system reset vector is at absolute address 0. All others are ORed with VBA as the base address.
Synchronous detection is acceptance of a request within an instruction cycle.
Asynchronous detection is acceptance of a request between instruction cycles or at any point in time after.

5.1 RST: System Reset

A system reset resets all states under all circumstances.

Type: Asynchronous hardware reset after RST pin level detection.

Vector address: Absolute address 0 (0x00000000).

Conditions: Non-maskable (unconditional)

PL after interrupt transition: 15

5.2 IAE: Instruction Access Exception

An instruction access exception is generated when an instruction is fetched from an undefined memory space. If the instruction is converted to a NOP by delayed instruction control (x-bit manipulation), then no exception will be generated.

Type: Instruction-synchronous exception caused by memory access error during instruction fetch.

Vector number/address: Vector number = 1 / VBA+0x010

Conditions: Non-maskable (unconditional). Invalidated by delayed instruction control (x-bit).

Saved address: Address of the instruction that caused the exception.

PL after interrupt transition: 15

5.3 IAIE: Instruction Address Invalid Exception

An instruction address invalid exception is generated when a register indirect branch instruction attempts an instruction fetch at an address that is not on a word boundary. If the instruction is converted to a NOP by delayed instruction control (x-bit manipulation), then no exception will be generated.

Type: Instruction-synchronous exception caused by an illegal JLR or RT instruction.

Vector number/address: Vector number = 2 / VBA+0x020

Conditions: Non-maskable (unconditional). Invalidated by delayed instruction control (x-bit).

Saved address: Address of the instruction that caused the exception.

PL after interrupt transition: 15

5.4 PIE: Privileged Instruction Exception

A privileged instruction exception is generated when an action that can only be performed in supervisor mode attempted in user mode: (a) in user mode a privileged register is specified as a destination, or (b) in user mode a number 64 or below is specified for a TRAP instruction vector. If the instruction is converted to a NOP by delayed instruction control (x-bit manipulation), then no exception will be generated.

Type: Instruction-synchronous exception caused by an illegal privileged instruction.

Vector number/address: Vector number = 4 / VBA+0x040

Conditions: Non-maskable (unconditional). Invalidated by delayed instruction control (x-bit).

Saved address: Address of the instruction that caused the exception.

PL after interrupt transition: 15

5.5 DAIE: Data Address Invalid Exception

A data address invalid exception is generated when a memory access instruction attempts to access a memory address not on a word boundary.

Type: Asynchronous exception caused by an illegal memory access instruction.

Vector number/address: Vector number = 5 / VBA+0x050

Conditions: EM == 1. However, exception must be maintained until accepted.

Saved address: Address being executed when the exception was accepted.

PL after interrupt transition: 15

5.6 DAE: Data Access Exception

A data access exception is generated when data is accessed in an undefined memory space.

Type: Asynchronous exception caused by a memory access instruction error.

Vector number/address: Vector number = 6 / VBA+0x060

Conditions: EM == 1. However, exception must be maintained until accepted.

Saved address: Address being executed when the exception was accepted.

PL after interrupt transition: 15

5.7 BPT: Breakpoint Trap

A breakpoint trap is generated when the instruction execution address matches the address pointed to by the %BPA register. However, the EBP bit in the %PSR register must be enabled. The instruction at the address that causes the trap will not be executed. The trap will be generated even if the instruction is converted to a NOP by delayed instruction control (x-bit manipulation).

Type: Instruction-synchronous trap caused by hardware.

Vector number/address: Vector number = 8 / VBA+0x080

Conditions: EM == 1 && EBP == 1. Not invalidated by delayed instruction control (x-bit).

Saved address: Address pointed to by the %BPA register.

PL after interrupt transition: 15

5.8 EINT: External Interrupt 1-15

External interrupts are generated by inputs. However, an external interrupt will be accepted only when its level has higher priority than the current processor level. When an external interrupt is accepted, the processor level becomes the same as its level.

Type: Asynchronous interrupt when level on INT1-INT15 pins is detected.

Vector number/address: Vector number = 33-47 / VBA+0x210-0x2f0

Conditions: EM == 1 && PL < external_interrupt_number

Saved address: Address being executed when the interrupt was accepted.

PL after interrupt transition: External interrupt number

The MSM7630 assigns interrupt levels as follows. It does not use other interrupts (including NMI).

Interrupt Source	Priority	Interrupt Number
User Block/TMR2	1	INT11
External pin (EXTINT)	2	INT9
Serial I/O	3	INT7
Parallel I/O	4	INT5
TMR1	5	INT3

5.9 Return From Interrupt

In order to return from an interrupt process caused by an exception, external interrupt, or software trap, the pipeline at the time of the interrupt must be regenerated before execution.

There are two types of returns: (1) re-execution of an instruction that was in its execution phase at the time an exception, external interrupt, or asynchronous trap caused an interrupt, and (2) re-execution of the instruction after the instruction that was in its execution phase at the time a software trap caused an interrupt. However, if breakpoints are supported by software traps then case (1) applies.

The return sequence from an interface process is described below.

Also, an rt instruction must not be executed while the EM bit of %PSR is 1 (the state permitting overlapping interrupts). If an interrupt occurred during the rt instruction in such a case, then the contents of %PSR would be corrupted.

1. Resume from interrupted instruction

```
    .
    .
brst    %psr, 4, %psr   ; EM-bit reset
jlr     %r1, %nop       ; delay slot, branch %r1 (old %PC),
                        ; return address not saved
rt      %r2             ; return to %r2 (old %nPC), %prPSR move to %PSR
```

2. Return from instruction after interrupt

```
    .
    .
add     %r2, 4, %r1     ; %r2+4 (old %nPC+4)→%r1
brst    %psr, 4, %psr   ; EM-bit reset
jlr     %r2, %nop       ; delay slot, branch %r2 (old %nPC), return address not saved
rt      %r1             ; return to %r1 (old %nPC+4), %prPSR move to %PSR
```

In this case the pNOP bit of %prPSR must be cleared in advance of rt instruction execution. If the pNOP bit of %prPSR is set and then the rt instruction is executed, then the instruction at the return point would not be executed.

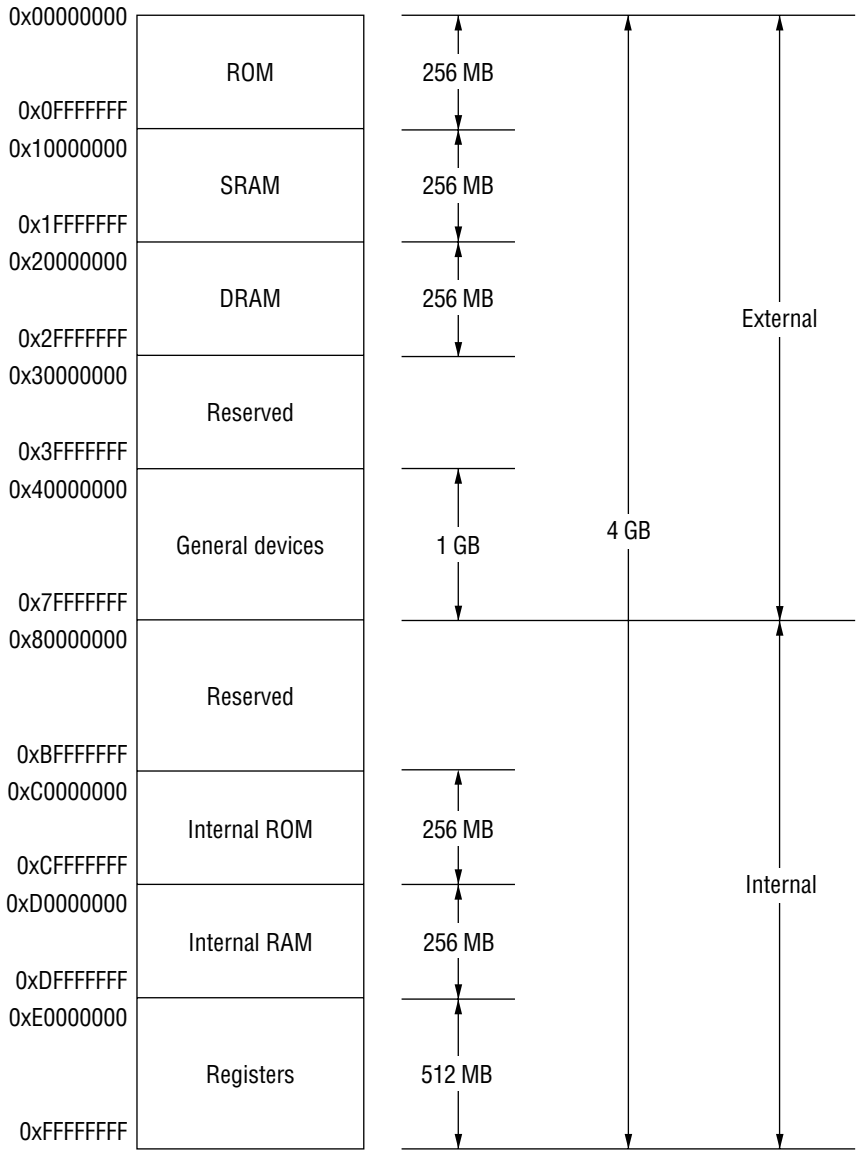
Bus Interface Unit

1. Features

The SCP's bus interface unit (BIU) manages address space and outputs control signals that enable optimal memory access. This allows ROM, SRAM, DRAM and other general devices to be accessed.

2. Address Space

The address space that can be directly accessed by load/store instructions is 4 gigabytes. The BIU manages this address space by dividing it into several.



2.1 ROM Space

ROM space is assigned to 0x00000000-0x0FFFFFFF. When this space is accessed the $\overline{\text{ROM}}$ signal goes "L".

2.2 SRAM Space

SRAM space is assigned to 0x10000000-0x1FFFFFFF. When this space is accessed the $\overline{\text{SRAM}}$ signal goes "L".

2.3 DRAM Space

DRAM space is assigned to 0x20000000-0x2FFFFFFF. When this space is accessed the DRAM controller outputs a signal required for DRAM access.

2.4 General Device Space

General device space is assigned to 0x40000000-0x7FFFFFFF. When this space is accessed the $\overline{\text{AS}}$ signal goes "L". This space is used to access general devices external to the MSM7630.

2.5 Internal ROM Space

Internal ROM space is assigned to 0xC0000000-0xCFFFFFFF. It is used to access internal ROM.

This space is not used by the MSM7630. Accesses to this space will cause instruction access exceptions or data access exceptions.

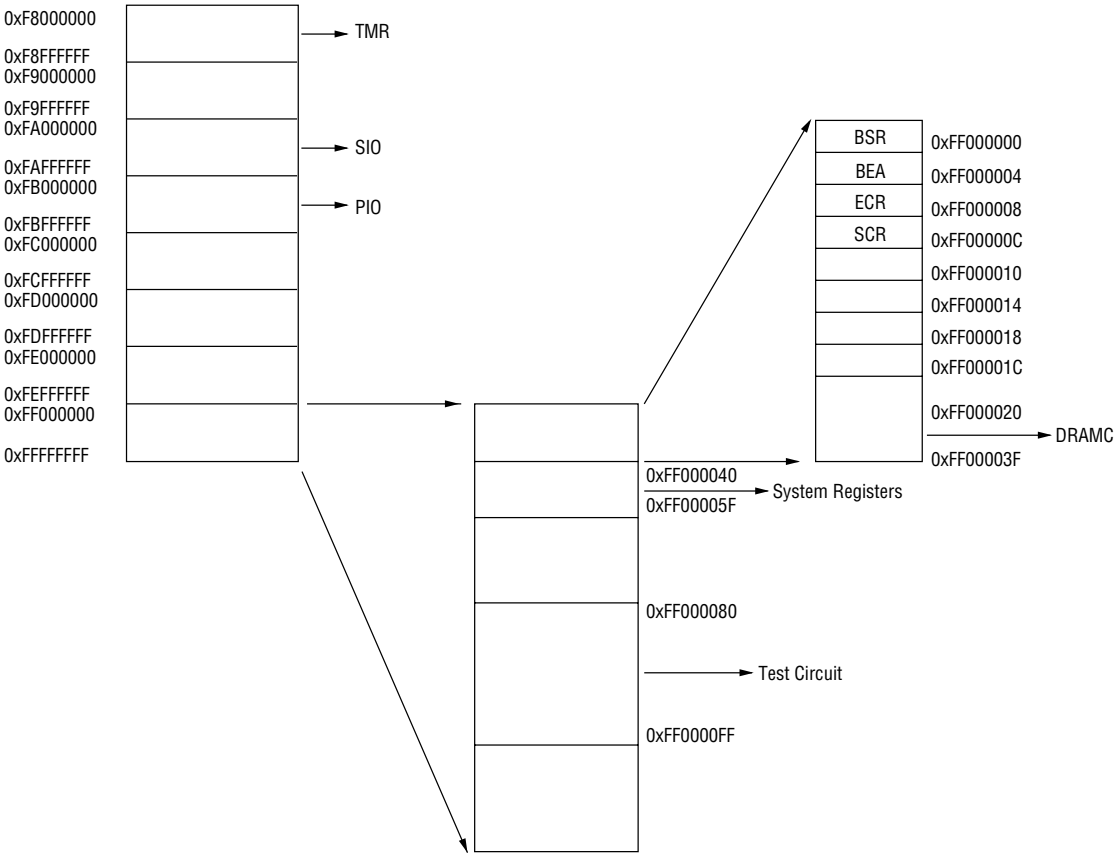
2.6 Internal RAM Space

Internal RAM space is assigned to 0xD0000000-0xDFFFFFFF. It is used to access internal RAM.

This space is not used by the MSM7630. Access to this will cause instruction access exceptions or data access exceptions.

2.7 Register Space

Register space is assigned to 0xE0000000-0xFFFFFFFF. Within this space, 0xF8000000-0xFFFFFFFF is assigned for standard I/O and system registers.

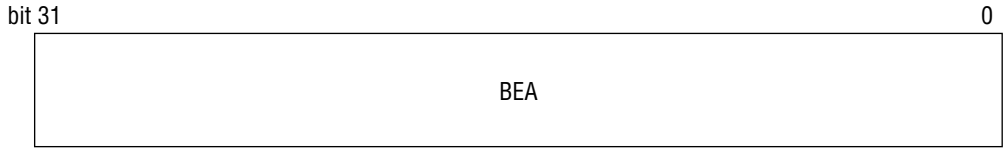

0xF8000000
0xF8FFFFFF
0xF9000000
0xF9FFFFFF
0xFA000000
0xFAFFFFFF
0xFB000000
0xFBFFFFFF
0xFC000000
0xFCFFFFFF
0xFD000000
0xFDFFFFFF
0xFE000000
0xFEFFFFFF
0xFF000000
0xFFFFFFFF
TMR
SIO
PIO
BSR
BEA
ECR
SCR
0xFF000000
0xFF000004
0xFF000008
0xFF00000C
0xFF000010
0xFF000014
0xFF000018
0xFF00001C
0xFF000020
0xFF00003F
DRAMC
0xFF000040
0xFF00005F
System Registers
0xFF000080
Test Circuit
0xFF0000FF

3. Registers

This is a register group used for bus control.

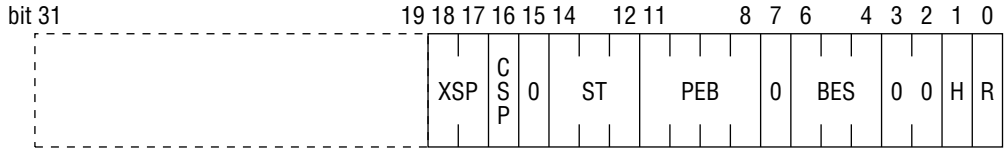
3.1 BEA: Bus Error Address

This register provides the address at the time a bus error occurred.



3.2 BSR: Bus Status Register

This register provides bus status information.



- bit[18:17] XSP: Sleep (read/write)

  When the $\overline{\text{STBY}}$ signal is "L", these bits either stop the clock without CPU intervention (XSP = 00) or stop the clock after waiting for the CPU suspend process (XSP = 11).

- bit[16] CSP: CPU Sleep (read/write)

  This bit indicates whether the CPU core is operating or suspended. Writing "1" will stop the CPU core's clock.

- bit[14:12] ST: Status (read-only)

  These bits save the status signals when an access by the CPU core causes a bus error.

- bit[11:8] PEB: Parity Error Byte (read-only)

  These bits provide the byte position when a parity error occurs.

- bit[6:4] BES: Bus Error Status (read-only)

  These bits provide the source of a bus error.

BES = 000	No error
BES = 001	BIU register privilege violation
BES = 010	Parity error
BES = 100	Invalid space access

  These bits will be "000" after reset.

• bit[1] H: Hold (read/write)

This bit sets whether or not bus rights will be passed upon a CPU core bus rights request. This bit will be "0" after reset.

3.3 ECR: Extra Configuration Register

This register sets bus operation.

• bit[10] OX: Internal ROM (read-only)

This bit indicates whether or not internal ROM will be accessed in 2 clocks.
MSM7630 does not use this bit.

• bit[9] AX: Internal RAM (read-only)

This bit indicates whether or not internal RAM will be accessed in 2 clocks.
MSM7630 does not use this bit.

• bit[7] PM: Parity Mode (read/write)

This bit sets parity.

PM = 0	Even parity
PM = 1	Odd parity

This bit will be "0" after reset.
MSM7630 does not use parity checking, so it ignores this field.

• bit[3] A: All Internal ROM (read/write)

This bit sets whether or not internal ROM will be accessed instead of external ROM.
MSM7630 has no internal ROM, so this bit is always "0".

• bit[2] OV: Internal ROM Valid (read-only)

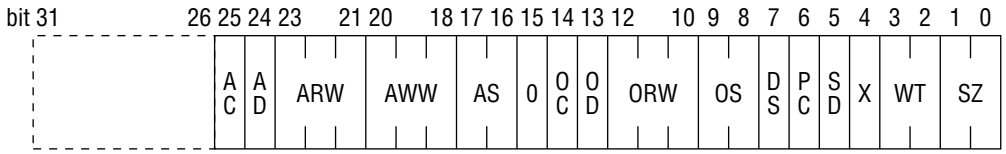
This bit shows whether internal ROM is enabled or disabled.
This bit is "0" for MSM7630.

• bit[1] AV: Internal RAM Valid (read-only)

This bit shows whether internal RAM is enabled or disabled.
This bit is "0" for MSM7630.

3.4 SCR: Space Configuration Register

This register sets ROM space, SRAM space, and general device space.



- bit[25] AC: SRAM Parity Check (read/write)

  This bit sets parity checking of SRAM space. It will be "0" after reset.

AC = 0	Ignore parity checks.
AC = 1	Generate a bus error if a parity error is detected.

- bit[24] AD: SRAM Dummy Cycle (read/write)

  This bit sets whether or not SRAM space may be accessed continuously after ROM space or DRAM space has been read.

AD = 0	Continuous access allowed.
AD = 1	Open an interval of at least one clock.

  This bit will be "1" after reset.

- bit[23:21] ARW: SRAM Read Wait (read/write)

  These bits set the wait count when SRAM space is accessed by a read.

ARW = 000	2τ access (1 wait)
ARW = 001	3τ access (2 waits)
ARW = 010	4τ access (3 waits)
ARW = 011	5τ access (4 waits)
ARW = 100	6τ access (5 waits)
ARW = 101	8τ access (7 waits)
ARW = 110	10τ access (9 waits)
ARW = 111	12τ access (11 waits)

  These bits will be "111" after reset.

• bit[20:18] AWW: SRAM Write Wait (read/write)

These bits set the wait count when SRAM space is accessed by a write.

AWW = 000	2τ access (1 wait)
AWW = 001	3τ access (2 waits)
AWW = 010	4τ access (3 waits)
AWW = 011	5τ access (4 waits)
AWW = 100	6τ access (5 waits)
AWW = 101	8τ access (7 waits)
AWW = 110	10τ access (9 waits)
AWW = 111	12τ access (11 waits)

These bits will be "111" after reset.

• bit[17:16] AS: SRAM Device Size (read/write)

These bits set the device size of SRAM space.

AS = 00	No SRAM (space is invalid)
AS = 01	8-bit wide device
AS = 10	16-bit wide device
AS = 11	32-bit wide device

These bits will be "00" after reset. When this field is "00", attempting to access SRAM space will cause an instruction access exception or data access exception.

• bit[14] OC: ROM Parity Check (read/write)

This bit sets parity checking for ROM space. It will be "0" after reset.

OC = 0	Ignore parity errors.
OC = 1	Generate a bus error if a parity error is detected.

This bit will be "0" for the MSM7630.

• bit[13] OD: ROM Dummy Cycle (read/write)

This bit sets whether or not a ROM space access will immediately follow an SRAM space or DRAM space read.

OD = 0	Consecutive access enabled.
OD = 1	Force an interval of at least one clock.

This bit will be "1" after reset.

• bit[12:10] ORW: ROM Read Wait (read/write)

These bits set the wait count when ROM space is accessed by a read.

ORW = 000	2τ access (1 wait)
ORW = 001	3τ access (2 waits)
ORW = 010	4τ access (3 waits)
ORW = 011	5τ access (4 waits)
ORW = 100	6τ access (5 waits)
ORW = 101	8τ access (7 waits)
ORW = 110	10τ access (9 waits)
ORW = 111	12τ access (11 waits)

These bits will be "111" after reset.

• bit[9:8] OS: ROM Device Size (read/write)

These bits set the device size of ROM space.

OS = 00	No ROM (space is invalid)
OS = 01	8-bit wide device
OS = 10	16-bit wide device
OS = 11	32-bit wide device

When this field is "00", attempting to access ROM space will cause an instruction access exception or data access exception.

• bit[7] DS: Other Data Setup (read/write)

This bit sets whether or not the data setup time to the write strobe signal $\overline{WR}$ is guaranteed during writes to general device space.

DS = 0	Not guaranteed.
DS = 1	Guaranteed.

This bit will be "1" after reset.

• bit[6] PC: Other Parity Check (read/ write)

This bit sets parity checking for general device space. It will be "0" after reset.

PC = 0	Ignore parity errors.
PC = 1	Generate a bus error if a parity error is detected.

This bit will be "0" for the MSM7630.

• bit[5] SD: Other Dummy Cycle (read/write)

This bit sets whether or not a general device space access will immediately follow an SRAM space or DRAM space read.

SD = 0	Consecutive access enabled.
SD = 1	Force an interval of at least one clock.

This bit will be "1" after reset.

• bit[4] X: External Bus Clock Unit (read/write)

This bit sets the operating clock unit for general device space.

X = 0	Use 1 clock as the unit.
X = 1	Use 2 clocks as the unit.

This bit will be "0" after reset.

• bit[3:2] WT: Other Wait (read/write)

These bits set the wait count when general device space is accessed.

WT = 00	4τ access
WT = 01	5τ access
WT = 10	6τ access
WT = 11	7τ access

These bits will be "11" after reset.

• bit[1:0] SZ: Other Device Size (read/write)

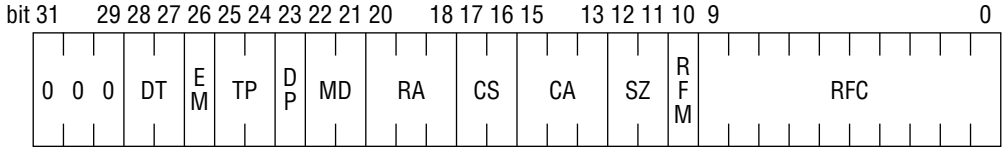
These bits set the device size of general device space.

SZ = 00	No general device (space is invalid)
SZ = 01	8-bit wide device
SZ = 10	16-bit wide device
SZ = 11	32-bit wide device

These bits will be "11" after reset. When this field is "00", attempting to access general device space will cause an instruction access exception or data access exception.

3.5 DRAM: DRAM Configuration Register

This register sets DRAM space.



After this register has been written, DRAM must not be accessed until the DRAM is operating properly. Refer to the data sheet of the DRAM used to obtain the required conditions for proper DRAM operation.

• bit[28:27] DT: Device Type (read/write)

These bits set the DRAM device type.

DT = 00	Fast page mode
DT = 01	Hyperpage mode (EDO DRAM)

These bits will be "00" after reset.

• bit[26] PR: Parity Check (read/write)

This bit sets parity checking for DRAM space. It will be "0" after reset.

PR = 0	Ignore parity errors.
PR = 1	Generate a bus error if a parity error is detected.

This bit will be "0" for the MSM7630.

• bit[25:24] TP: Type (read/write)

This bit sets the DRAM's RAS signal and byte position control signal.

TP = 00	1 RAS mode, byte position CAS control
TP = 01	2 RAS mode, byte position CAS control
TP = 10	1 RAS mode, byte position WE control
TP = 11	2 RAS mode, byte position WE control

These bits will be "00" after reset.

• bit[23] DP: Data Priority (read/write)

This bit sets the priority of processing when data access is requested by a load/store instruction during a one-line instruction cache read from DRAM due to an instruction cache miss.

DP = 0	Give priority to the instruction cache read from DRAM.
DP = 1	Give priority to the data access.

This bit will be "0" after reset.

• bit[22:21] MD: Mode (read/write)

These bits set the number of clocks for a DRAM access.

MD = 01	2n clock access
MD = 10	3n clock access

These bits will be "10" after reset.

• bit[20:18] RA: Row Address (read/write)

These bits set the most significant bit position of the row address.

RA = 000	A17
RA = 001	A18
RA = 010	A19
RA = 011	A20
RA = 100	A21
RA = 101	A22
RA = 110	A23

These bits will be "000" after reset.

• bit[17:16] RS: Row Shift (read/write)

These bits set how many bits to shift the row address to output it as a DRAM address.

RS = 00	8-bit shift
RS = 01	9-bit shift
RS = 10	10-bit shift
RS = 11	11-bit shift

These bits will be "00" after reset.

• bit[15:13] CA: Column Address (read/write)

These bits set the most significant bit position of the column address.

CA = 000	A08
CA = 001	A09
CA = 010	A10
CA = 011	A11
CA = 100	A12

These bits will be "000" after reset.

- bit[12:11] SZ: Device Size (read/write)

  These bits set the device size of DRAM space.

SZ = 00	No DRAM (space is invalid)
SZ = 01	8-bit wide device
SZ = 10	16-bit wide device
SZ = 11	32-bit wide device

  These bits will be "00" after reset. When this field is "00", attempting to access DRAM space will cause an instruction access exception or data access exception.

- bit[10] RFM: Refresh Mode (read/write)

  This bit sets the refresh operation mode.

RFM = 0	CAS-before-RAS refresh
RFM = 1	CAS-before-RAS self-refresh

  This bit will be "0" after reset.

- bit[9:0] RFC: Refresh Counter (read/write)

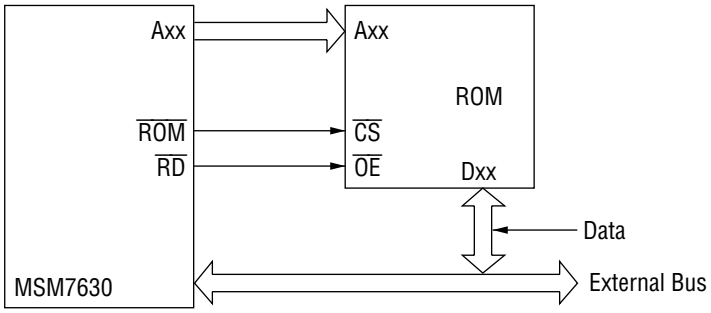
  These bits set the initial value of the refresh counter. It should be set as an integer value obtained by:

  [(refresh period) ÷ (clock period) ÷ 16] – 1

  These bits will be "0000000000" after reset.

4. ROM Access

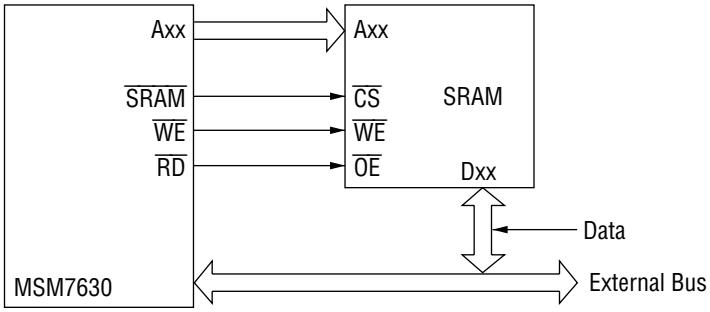
The MSM7630 interface with ROM is shown below.



The $\overline{\text{ROM}}$ signal will become "0" when the address signal and specified ROM space match. Refer to the timing diagram for basic timing of ROM accesses.

5. SRAM Access

The MSM7630 interface with SRAM is shown below.

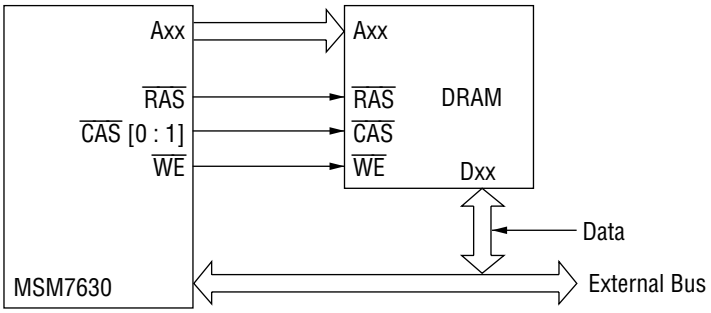


The $\overline{\text{SRAM}}$ signal will become "0" when the address signal and specified SRAM space match. Refer to the timing diagram for basic timing of SRAM accesses.

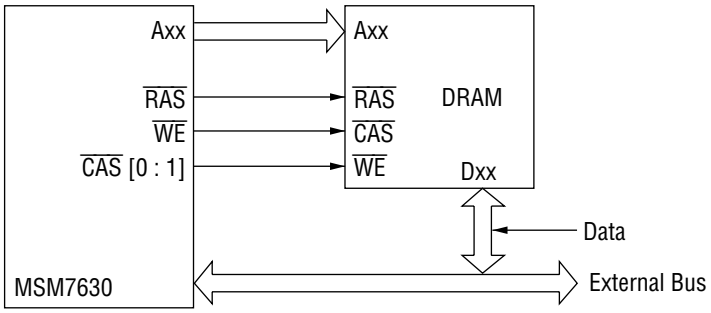
6. DRAM Access

There are two MSM7630 interfaces with DRAM: one when byte position is specified by $\overline{CAS}$, and one when byte position is specified by $\overline{WE}$. This is set by the DRAM register's TP field.

An interface example when byte position is specified by $\overline{CAS}$ is shown below.



An interface example when byte position is specified by $\overline{WE}$ is shown below.



Refer to the timing chart for basic timing of DRAM accesses.

The table below shows how address signals are connected for different DRAM configurations.

Configuration	Row	Column	Address lines	CA	RS	RA
256K × 8	17-09	08-00	A[08:00]	000	01	000
256K × 16	18-10	09-01	A[09:01]	001	01	001
	18-09	08-01	A[09:01]	000	00	001
256K × 32	19-11	10-02	A[10:02]	010	01	010
	19-10	09-02	A[11:02]	001	00	010
512K × 8	18-09	08-00	A[09:00]	000	01	001
512K × 16	19-10	09-01	A[10:01]	001	01	010
512K × 32	20-11	10-02	A[11:02]	010	01	011
1M × 8	19-10	09-00	A[09:00]	001	10	010
1M × 16	20-11	10-01	A[10:01]	010	10	011
	20-10	09-01	A[11:01]	001	01	011
	20-09	08-01	A[12:01]	000	00	011
1M × 32	21-12	11-02	A[11:02]	011	10	100
	21-11	10-02	A[12:02]	010	01	100
	21-10	09-02	A[13:02]	001	00	100
2M × 8	20-10	09-00	A[10:00]	001	10	011
	20-09	08-00	A[11:00]	000	01	011
2M × 16	21-11	10-01	A[11:01]	010	10	100
	21-10	09-01	A[12:01]	001	01	100
2M × 32	22-12	11-02	A[12:02]	011	10	101
	22-11	10-02	A[13:02]	010	01	101
4M × 8	21-11	10-00	A[10:00]	010	11	100
	21-10	09-00	A[11:00]	001	10	100
4M × 16	22-12	11-01	A[11:01]	011	11	101
	22-11	10-01	A[12:01]	010	10	101
4M × 32	23-13	12-02	A[12:02]	100	11	110
	23-12	11-02	A[13:02]	011	10	110

Serial Interface

1. Features

The serial interface (SIO) performs both clock synchronized and start-stop transfers.

2. SIO Functions

2.1 Port Configuration

- Independent transmit and receive circuits
- Double buffer configuration for receive buffer

Because the transmit and receive circuits are independent, start-stop transfers are all full-duplex communication.

2.2 Transfer Methods

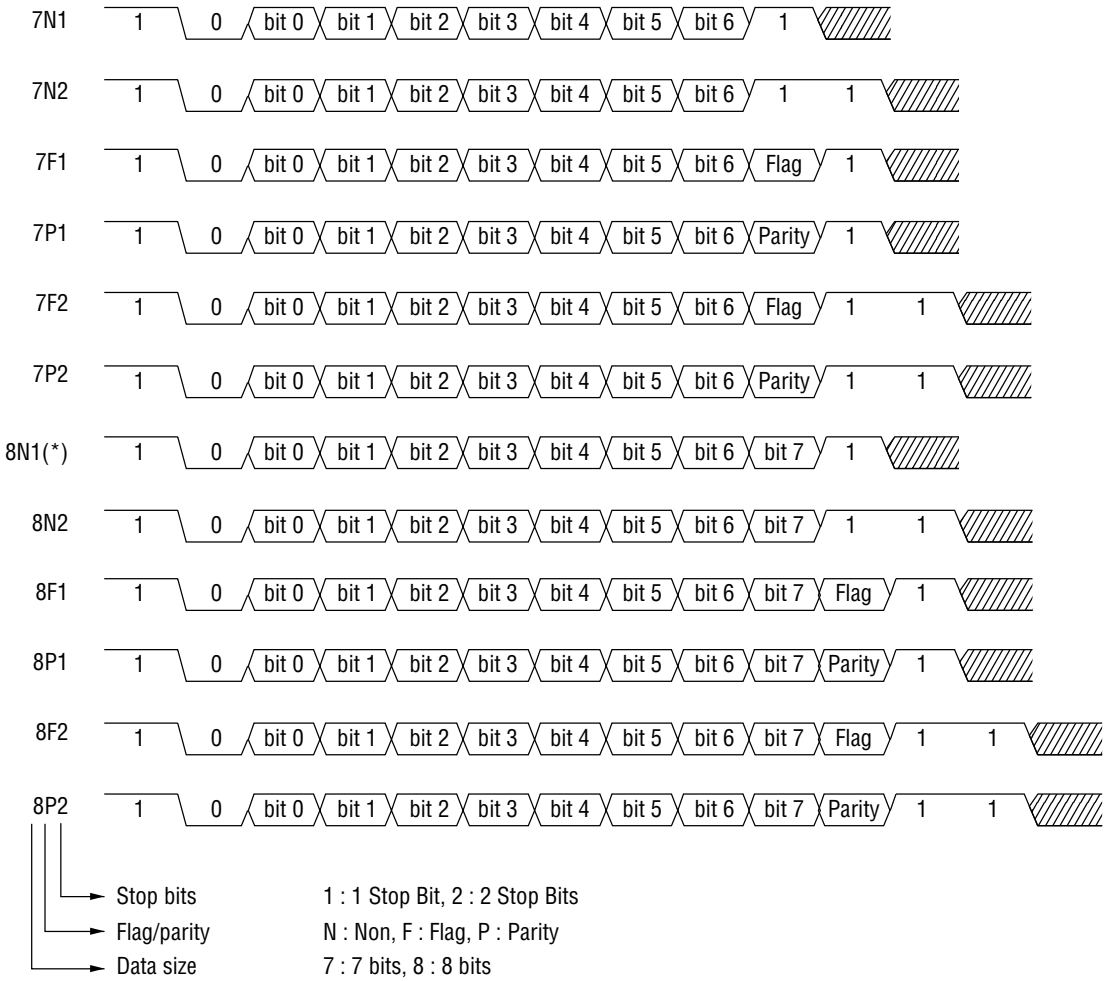
- Start-stop transfer

Data length:	7 bits or 8 bits selectable
Transfer sequence:	LSB first
Stop bits:	1 bit or 2 bits selectable
Parity bit:	No parity, even parity, or odd parity selectable
Flag bit:	Enables inter-processor communication using the serial port. However, cannot be used together with parity bit.

- Clock synchronized transfer

Data length:	8 bits fixed
Transfer sequence:	LSB first

The chart below shows the data format with start-stop transfers.



(*) After reset the format will be 8 bits, no parity, 1 stop bit.

2.3 Baud Rate

- Internal baud rate generator

- Clock synchronized transfers

$B = f/(8 \times n \times (256 - P))$

where

B : baud rate (bps)
f : processor (SCP) clock frequency (Hz)
n : baud rate parameter
One of 1, 2, 4, 8, 16, 32, and 64. Selected by SBR's SBRP field.
(Refer to the register description.)
P : baud rate adjustment value ($0 \leq P \leq 255$)
Set by SBR's SBRV field. (Refer to the register description.)

At a processor (SCP) clock of 20 MHz, the maximum transfer rate is 2.5 Mbps. At 40 MHz, the maximum transfer rate is 5 Mbps.

- Start-stop transfers

$B = f/(16 \times n \times (256 - SBR))$

where

B : baud rate (bps)
f : processor (SCP) clock frequency (Hz)
n : baud rate parameter
One of 1, 2, 4, 8, 16, 32, and 64. Selected by SBR's SBRP field.
(Refer to the register description.)
SBR : baud rate adjustment value ($0 \leq SBR \leq 255$)
Set by SBR's SBRV field. (Refer to the register description.)

2.4 Error Detection

- Parity errors (start-stop transfers)

A parity error will be detected when a parity bit generated from received data does not match the received parity bit.

- Framing errors (start-stop transfers)

A framing error will be detected when a received stop bit is "0". When 2 stop bits have been selected, only the first bit received will be checked.

- Overrun errors (start-stop and clock synchronized transfers)

An overrun error will be detected when the next receive frame's stop bit is detected before the receive buffer has been read.

2.5 Interrupts

The SIO is the source of the following interrupts.

- Interrupts

  - Receive error interrupt
    A receive error interrupt will be generated whenever a parity error, framing error, or overrun error is detected.

  - Receive buffer full interrupt
    A receive buffer full interrupt will be generated whenever the valid receive data has been transferred to the receive buffer.

  - Transmit buffer empty interrupt
    A transmit buffer empty interrupt will be generated whenever the transmit buffer becomes empty.

  - Transmit end interrupt
    A transmit end interrupt will be generated whenever an SIO data transfer ends.

  - Modem status interrupt
    A modem status interrupt will be generated whenever a change in a modem control input signal (CTS, DSR) is detected.

- Interrupt enable/disable

  Each interrupt source can be independently enabled or disabled. Also, all interrupts can be disabled at once.

- Interrupt requests

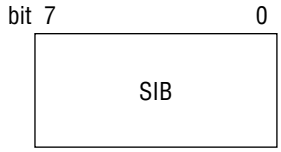
  Whenever any of the five interrupts above is enabled and its conditions are fulfilled, the CPU will get an SIO interrupt request.

3. SIO Registers

These registers control SIO.

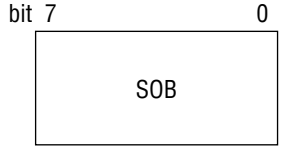
3.1 SIB: SIO Input Buffer

This register holds data that has been input externally. It is undefined after reset.



3.2 SOB: SIO Output Buffer

This register holds data to be output externally. It is undefined after reset.



3.3 SSTS: SIO Status Register

This register provides SIO status.

bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	SOVE	SFRE	SPTE	0	SMSI	SOST	SIST	SERI	SRXI	STXI	STEI

• bit[10] SOVE

0 : No overrun error
1 : Overrun error generated

This bit can only be written with "0". It will be "0" after reset.

• bit[9] SFRE

0 : No framing error
1 : Framing error generated

This bit can only be written with "0". It will be "0" after reset.

• bit[8] SPTE

0 : No parity error
1 : Parity error generated

This bit can only be written with "0". It will be "0" after reset.

• bit[6] SMSI

0 : No modem status interrupt
1 : Modem status interrupt requested

This bit is read-only. It will be "0" after reset.

• bit[5] SOST

0 : Transmit buffer full
1 : Transmit buffer empty

This bit can only be written with "0". It will be "1" after reset.

• bit[4] SIST

0 : Receive buffer empty
1 : Receive buffer full

This bit can only be written with "0". It will be "0" after reset.

• bit[3] SERI

0 : No receive error interrupt
1 : Receive error interrupt requested

This bit is read-only. It will be "0" after reset.

• bit[2] SRXI

0 : No receive buffer full interrupt
1 : Receive buffer full interrupt requested

This bit is read-only. It will be "0" after reset.

• bit[1] STXI

0 : No transmit buffer empty interrupt
1 : Transmit buffer empty interrupt requested

This bit is read-only. It will be "0" after reset.

• bit[0] STEI

0 : No transmit end interrupt
1 : Transmit end interrupt requested

This bit can only be written with "0". It will be "0" after reset.

3.4 SCMD: SIO Command Register

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SREN	STEN	0	SIEN	SERIE	SRXIE	STXIE	STEIE	0	SMOD	SFBM	SFB	SFL	SPTY		SSTP

• bit[15] SREN

0 : Data receive disabled
1 : Data receive enabled

This bit will be "0" after reset.

• bit[14] STEN

0 : Data transmit disabled
1 : Data transmit enabled

This bit will be "0" after reset.

• bit[12] SIEN

0 : Interrupts disabled
1 : Interrupts enabled

This bit will be "0" after reset.

• bit[11] SERIE

0 : Receive error interrupts disabled
1 : Receive error interrupts enabled

This bit will be "0" after reset.

• bit[10] SRXIE

0 : Receive buffer full interrupts disabled
1 : Receive buffer full interrupts enabled

This bit will be "0" after reset.

• bit[9] STXIE

0 : Transmit buffer empty interrupts disabled
1 : Transmit buffer empty interrupts enabled

This bit will be "0" after reset.

• bit[8] STEIE

0 : Transmit end interrupts disabled
1 : Transmit end interrupts enabled

This bit will be "0" after reset.

• bit[6] SMOD

0 : Start-stop transfer mode
1 : Clock synchronized transfer mode

This bit will be "0" after reset.

• bit[5] SFBM

0 : Clear flag bit mode
1 : Set flag bit mode

This bit will be "0" after reset.

• bit[4] SFB

0 : Flag bit value set to "0"
1 : Flag bit value set to "1"

This bit will be "0" after reset.

• bit[3] SFL

0 : Transfer data length is 8 bits
1 : Transfer data length is 7 bits

This bit will be "0" after reset.

• bit[2:1] SPTY

00 : No parity
10 : Even parity
11 : Odd parity

These bits will be "00" after reset.

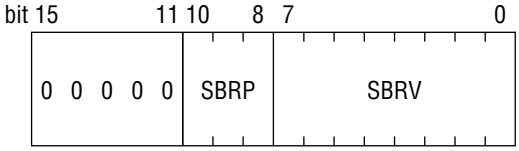
• bit[0] SSTP

0 : 1 stop bit
1 : 2 stop bits

This bit will be "0" after reset.

3.5 SBR: Baud Rate Adjustment Register

This register sets values that adjust the baud rate.



• bit[10:8] SBRP

000 : Baud rate parameter n = 1
001 : Baud rate parameter n = 2
010 : Baud rate parameter n = 4
011 : Baud rate parameter n = 8
100 : Baud rate parameter n = 16
101 : Baud rate parameter n = 32
110 : Baud rate parameter n = 64

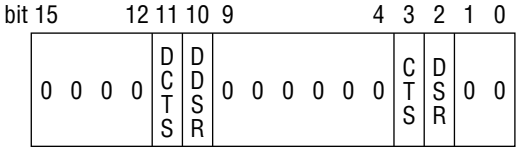
These bits will be undefined after reset.

• bit[7:0] SBRV

These bits are the baud rate adjustment value. They will be undefined after reset.

3.6 MSTS: Modem Status Register

This register provides the states of modem signals.



• bit[11] DCTS

0 : CTS signal has not changed
1 : CTS signal has changed

This bit is read-only. It will be "0" after reset.

• bit[10] DDSR

0 : DSR signal has not changed
1 : DSR signal has changed

This bit is read-only. It will be "0" after reset.

• bit[3] CTS

0 : CTS input signal value = "0"
1 : CTS input signal value = "1"

This bit is read-only. It will be the CTS pin input value after reset.

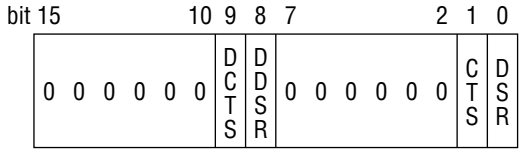
• bit[2] DSR

0 : DSR input signal value = "0"
1 : DSR input signal value = "1"

This bit is read-only. It will be the DSR pin input value after reset.

3.7 MCMD: Modem Command Register

This register enables/disables modem status interrupts and auto-enable mode, and controls RTS and DTR output signals.



• bit[9] SMSIE

0 : Disables modem status interrupts
1 : Enables modem status interrupts

This bit will be "0" after reset.

• bit[8] SAEN

0 : Disables auto-enable mode
1 : Enables auto-enable mode

This bit will be "0" after reset.

• bit[1] RTS

0 : Output RTS signal "0"
1 : Output RTS signal "1"

This bit will be "0" after reset.

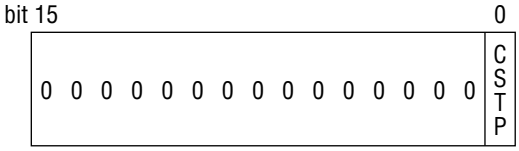
• bit[0] DTR

0 : Output DTR signal "0"
1 : Output DTR signal "1"

This bit will be "0" after reset.

3.8 SCNT: SIO Control Register

This register controls SIO.



• bit[0] CSTP

0 : Enable SIO clock supply
1 : Disable SIO clock supply

This bit will be "0" after reset.

4. SIO Register Addresses

SIO register addresses for the MSM7630 are shown below.

0xFA000000	SIO Input Buffer
0xFA000004	SIO Output Buffer
0xFA000008	Baud Rate Adjustment Register
0xFA00000C	SIO Status Register
0xFA000010	SIO Command Register
0xFA000014	Modem Status Register
0xFA000018	Modem Command Register
0xFA00001C	SIO Control Register

5. SIO Operation

There are two methods of SIO operation: start-stop transfers where communication is performed synchronized to characters, and clock synchronized transfers where communication is performed synchronized to the clock.

5.1 Clock Synchronized Transfers

Clock synchronized transfer mode is selected by setting the SCMD (Command Register) SMOD bit to "1". In this mode 8-bit data will be input/output synchronized to the clock output from the SCLK pin.

With clock synchronized transfers, transfer data is only 8 bits, so parity bits and flag bits cannot be added. The SCMD (Command Register) SFBM bit, SFL bit, and SPTY bits will be set to "0", "0", and "00" respectively.

5.1.1 Clock Synchronized Transfer Baud Rate

$$B = \frac{f}{8 \times n \times (256 - P)}$$

where

B : baud rate
f : SCP clock frequency
n : baud rate parameter (set by SBR register's SBRP bit)
P : baud rate adjustment value (set by SBR register's SBRV bit)

Set SBR (Baud Rate Adjustment Register) to achieve the required baud rate.

5.1.2 Clock Synchronized Transmit Operation

1) Verify that the SSTS (Status Register) SOST bit is "1", and then write the data to be transferred to the transmit buffer SOB.

2) Write "0" to SOST to indicate that SOB has valid data.

3) If using SIO interrupts, set the SCMD (Command Register) SIEN bit to "1". If using the transmit buffer empty interrupt, write "1" to the SCMD STXIE bit. If using the transmit end interrupt, write "1" to the SCMD STEIE bit.

4) If the MCMD (Modem Command Register) SAEN bit is "0", then setting the SCMD (Command Register) STEN bit to "1" will start the transfer. If the MCMD SAEN bit is "1", then the transfer will start when the SCMD STEN bit is "1" and the CTS input is "1".

5) SOB (Transmit Buffer) data will be transferred LSB first from the TXD output. Also, a synchronous clock will be transmitted from the SCLK pin. Data on the TXD output will change synchronous to the falling edge of SCLK. The receiving device should sample TXD data on the rising edge of SCLK.

6) When the next data can be written to the transmit buffer, the SSTS (Status Register) SOST bit will change from "0" to "1". If the SCMD (Command Register) STXIE and SIEN bits are "1" at this time, then the SSTS STXI bit will become "1" and an interrupt request to the CPU will be generated.

7) For continuous transfers, after the SSTS (Status Register) SOST bit becomes "1" write new data to SOB (Transmit Buffer) and write "0" to the SOST bit.

8) If there is no more data to be transmitted, then write "0" to the SCMD (Command Register) STXIE bit. This will disable interrupt requests from SIO.

9) When transfer of the eighth bit of data ends, the SSTS (Status Register) SOST bit will become "1" (transmit buffer SOB is empty), SCLK will stop, and the transmit operation will end. If the SCMD's STEIE and SIEN bits are "1" at this time, then the SSTS's STEI bit will become "1" and an interrupt request to the CPU will be generated. This interrupt can be released by writing "0" to the SSTS's STEI bit or the SCMD's STEIE bit.

5.1.3 Clock Synchronized Receive Operation

1) The receive operation will begin if the MCMD's SAEN bit is "0" (auto-enable mode disabled) and the SCMD's SREN bit is "1" (data receive enabled).

2) When the receive operation begins, a synchronous clock will be output from SCLK.

3) If using SIO interrupts, set SCMD's SIEN bit to "1". If using the receive buffer full interrupt, set SCMD's SRXIE bit to "1".

4) The transmitting device should input the data to be transferred on RXD on the falling edge of SCLK, LSB first. The SIO will sample RXD data on SCLK's rising edge, shifting it into the Receive Shift Register.

5) When the eighth bit of data has been received, the receive shift register's data is transferred to SIB. However, it will not be transferred to SIB if an overrun error occurs.

6) After data has transferred from the receive shift register to the receive buffer SIB, SIST will change from "0" to "1", indicating that there is valid data in the receive buffer SIB. If SCMD's SRXI bit and SIEN bit are both "1" at this time, an interrupt request to the CPU will be generated.

7) To continue receiving data, read the SIB data after SIS becomes "1", and then write "0" to SIST.

8) To end the receive operation, write "0" to SCMD's SREN bit. At the time "0" is written to SREN, data currently being received will be transferred to SIB and the receive operation will end.

9) When SSTS's SIST bit is "1" and the SIO enters the state in which data is ready to be transferred from the receive shift register to the receive buffer, the SIO will assume that an overrun error (receipt of further data before the value of the receive buffer SIB is read) has occurred. SSTS's SOVE bit will then be set to "1". In this case the receive shift register value will not be transferred to the receive buffer SIB. If SCMD's SERIE bit and SIEN bit are "1", then SSTS's SERI bit will be set to "1" and an interrupt request to the CPU will be generated. To release the interrupt, write "0" to SSTS's SOVE bit or to SCMD's SERIE or SIEN bit.

5.2 Start-Stop Transfers

Start-stop transfer mode is selected by setting the SCMD (Command Register) SMOD bit to "0". In this mode data is output LSB first from TXD, and input LSB first from RXD.

5.2.1 Start-Stop Transfer Baud Rate

$$B = \frac{f}{16 \times n \times (256 - P)}$$

where

B : baud rate
f : SCP clock frequency
n : baud rate parameter (set by SBR register's SBRP bit)
P : baud rate adjustment value (set by SBR register's SBRV bit)

Set SBR (Baud Rate Adjustment Register) to achieve the required baud rate.

5.2.2 Start-Stop Transmit Operation

1) Verify that the SSTS (Status Register) SOST bit is "1", and then write the data to be transferred to the transmit buffer SOB. Next write "0" to SOST to indicate that SOB has valid data.

2) If the MCMD (Modem Command Register) SAEN bit is "0", then setting the SCMD (Command Register) STEN bit to "1" will start the transfer. If the MCMD SAEN bit is "1", then the transfer will start when the SCMD STEN bit is "1" and the CTS input is "1".

3) For start-stop transmit operation, a start bit "0" will be output from TXD. Then the data written in SOB will be output LSB first. If SCMD's SFL bit is "0", then 8 bits of data will be output. If the SFL bit is "1", then 7 bits will be output.

4) When SCMD (Command Register) SFBM bit is "0", a parity bit will be output after the SOB data. The parity will be even if the SPTY field is "10", and odd if the SPTY field is "11". If SCMD's SFBM bit is "1" and SPTY is "00", then the value set in SCMD's SFB bit will be output after the SOB data. If SCMD's SFBM bit is "0" and the SPTY field is "0", then neither a parity bit nor flag bit will be output after SOB data.

5) Finally, one stop bit will be output if the SCMD (Command Register) SSTP bit is "0", or two stop bits will be output if the SSTP bit is "1". This will end the transfer of one frame of data.

6) When the next data can be written to the transmit buffer, the SSTS (Status Register) SOST bit will change from "0" to "1". If the SCMD (Command Register) STXIE and SIEN bits are "1" at this time, then the SSTS STXI bit will become "1" and an interrupt request to the CPU will be generated.

7) For continuous transfers, after the SSTS (Status Register) SOST bit becomes "1" write new data to SOB (Transmit Buffer) and write "0" to the SOST bit. This will disable interrupt requests from SIO.

8) If there is no more data to be transmitted, then write "0" to the SCMD (Command Register) STXIE bit. The will disable interrupt requests from SIO.

9) When transfer of the stop bit ends, the transmit operation will end if the SOST bit is "1". If the SCMD's STEIE and SIEN bits are "1" at this time, then the SSTS's STEI bit will become "1" and an interrupt request to the CPU will be generated. This interrupt can be released by writing "0" to the SSTS's STEI bit or the SCMD's STEIE or SINT bit.

5.2.3 Start-Stop Receive Operation

1) The receive operation can begin if the MCMD's SAEN bit is "0" and the SCMD's SREN bit is "1".

2) If using SIO interrupts, set SCMD's SIEN bit to "1". If using the receive buffer full interrupt, set SCMD's SRXIE bit to "1". If using the receive error interrupt, set SCMD's SERIE bit to "1".

3) The SIO receive operation will start when a falling edge is detected on RXD. The first bit of data is received as the start bit. If the received value is "1", then it will not be recognized as a start bit, the receive operation will be suspended, and the device will wait for another RXD falling edge to be detected. If the received value is "0", then data will continue to be received.

4) When the start bit is received, receive of data will start. If SCMD's SFL bit is "0", then 8 bits of data will be input serially into the receive shift register. If the SFL bit is "1", then 7 bits of data will be input.

5) When SCMD (Command Register) SFBM bit is "0", a parity bit will be received after the data. The parity will be even if the SPTY field is "10", and odd if the SPTY field is "11". If SCMD's SFBM bit is "1" and SPTY is "00", one flag bit will be received. If SCMD's SFBM bit is "0" and the SPTY field is "00", then neither a parity bit nor flag bit will be received.

6) Finally one stop bit will be received. Even if SCMD's SSTP bit is "1", only the first stop bit will be received.

7) When all bits have been received, the data input in the receive shift register will be transferred to the receive buffer SIB. However, if either of the following two conditions applies, then data will not be transferred to SIB, and SIB will retain its previous value.

   1. SCMD's SFBM bit is "1", its SPTY field is "00", and the received flag bit does not match SCMD's SFB bit.

   2. An overrun error occurred.

8) When data has been transferred from the receive shift register to the receive buffer SIB, SSTS's SIST will change from "0" to "1". If SCMD's SRXIE and SIEN bits are both "1", then SSTS's SRXI bit will become "1" and an interrupt request to the CPU will be generated. To release the interrupt, write "0" to SSTS's SIST bit or to SCMD's SRXIE or SIEN bit.

9) When SSTS's SIST bit is "1" and the SIO enters the state in which data is ready to be transferred from the receive shift register to the receive buffer, the SIO will assume that an overrun error (receipt of further data before the value of the receive buffer is read) has occurred. SSTS's SOVE bit will then be set to "1".

10) If the received stop bit is "0", then it will be considered indication of a framing error. SSTS's SFRE bit will then be set to "1".

11) When SCMD's SPTY field is "10" or "11", a mismatch between parity generated from the receive data and the parity bit will be considered a parity error. SSTS's SPTE bit will then be set to "1".

12) If one or more of the SOVE, SFRE, and SPTE bits are "1" and the SCMD' s SERIE and SIEN bits are "1", then SSTS's SERI bit will be set to "1" and an interrupt request to the CPU will be generated.

13) To release the interrupt for any error, write "0" to all of SSTS's SOVE, SFRE, and SPTE bits, or write "0" to SCMD's SERIE or SIEN bits.

Parallel Interface

1. Features

The parallel interface (PIO) inputs and outputs 8-bit wide parallel data. It has three data transfer methods: software control mode where input/output is specified with 1-bit ports, handshake control mode through strobe/acknowledge signals and flags indicating buffer status, and bus control mode through read/write signals.

2. PIO Functions

2.1 PIO Data Size

- 8 bits

2.2 PIO Control Modes

- Software control mode

In software control mode, the PIO controls input and output of bits in accordance with the value written in the direction register. If a direction register bit is "0", then the corresponding pin level will be an input. If "1", then the value in the corresponding output buffer will be output to the pin.

- Handshake control mode

In handshake control mode, the PIO inputs external data through a handshake using a strobe signal ($\overline{PSTB}$) and input buffer full signal (PIBF). It outputs data externally through a handshake using an output buffer full signal (POBF) and acknowledge signal ($\overline{PACK}$).

- Bus control mode

In bus control mode, the PIO controls data input/output with a chip select signal ($\overline{PCS}$), flag/buffer select signal (PIOA), read signal ($\overline{PACK}$), and write signal ($\overline{PSTB}$).

2.3 PIO Interrupts

Interrupts to the CPU core are available when handshake control mode or bus control mode is selected.

- Input buffer full interrupts

When PCMD's PIEN bit is "1", writing "0" to the PIIE bit will disable input buffer full interrupts, and writing "1" will enable them. When the PIEN bit is "0", input buffer full interrupts will be disabled regardless of the value of the PIIE bit.

If input buffer full interrupts are enabled, then one will be generated whenever the input buffer is written from an external device.

To release input buffer full interrupts, write "0" to the status register PSTS's PIST bit, to the command register PCMD's PIIE bit, or to PCMD's PIEN bit.

• Output buffer empty interrupts

When PCMD's POEN bit is "1", writing "0" to the POIE bit will disable output buffer empty interrupts, and writing "1" will enable them. When the POEN bit is "0", output buffer empty interrupts will be disabled regardless of the value of the POIE bit.

If output buffer empty interrupts are enabled, then one will be generated whenever the output buffer is read by an external device.

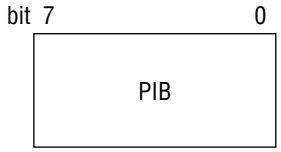
To release output buffer empty interrupts, write "0" to the status register PSTS's POST bit, to the command register PCMD's POIE bit, or to PCMD's POEN bit.

3. PIO Registers

These registers control the PIO.

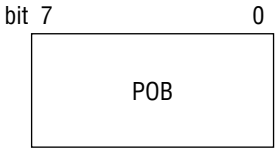
3.1 PIB: PIO Input Buffer

This buffer saves data input from an external device. It will be undefined after reset.



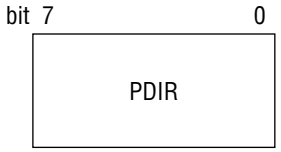
3.2 POB: PIO Output Buffer

This buffer saves data to be output to an external device. It will be undefined after reset.



3.3 PDIR: PIO Direction Register

This register specifies under software control whether each parallel port bit is input or output. It will be "00000000" after reset.



3.4 PSTS: PIO Status Register

This register provides the PIO status.

bit	7	6	5	4	3	2	1	0
	0	0	PINTI	PINTO	PACK	POST	PSTB	PIST

- bit[5] PINTI

  0 : No input buffer full interrupt
  1 : Input buffer full interrupt occurred

  This bit will be "0" after reset.

- bit[4] PINTO

  0 : No output buffer empty interrupts
  1 : Output buffer empty interrupt occurred

  This bit will be "0" after reset.

- bit[3] PACK

  0 : No acknowledge
  1 : Acknowledge

  This bit will be undefined after reset.

- bit[2] POST

  0 : Output buffer full
  1 : Output buffer empty

  This bit can only be written with "0". It will be "1" after reset.

- bit[1] PSTB

  0 : No strobe
  1 : Strobe

  This bit will be undefined after reset.

- bit[0] PIST

  0 : Input buffer empty
  1 : Input buffer full

  This bit can only be written with "0". It will be "0" after reset.

3.5 PCMD: PIO Command Register

This register specifies the parallel port mode and specifies whether interrupts are enabled or disabled. It will be "00000000" after reset.

bit 7	6	5	4	3	2	1	0
PMOD		POEN	PIEN	0	0	POIE	PIIE

- bit[7:6] PMOD
  00 : Bus control mode
  01 : Handshake control mode
  1x : Software control mode

- bit[5] POEN

  0 : Output operation disabled
  1 : Output operation enabled

- bit[4] PIEN

  0 : Input operation disabled
  1 : Input operation enabled

- bit[1] POIE

  0 : Output buffer empty interrupts disabled
  1 : Output buffer empty interrupts enabled

- bit[0] PIIE

  0 : Input buffer full interrupts disabled
  1 : Input buffer full interrupts enabled

4. PIO Register Addresses

For the MSM7630, PIO register addresses are listed below.

0xFB000000	PIO Input Buffer
0xFB000004	PIO Output Buffer
0xFB000008	PIO Direction Register
0xFB00000C	PIO Status Register
0xFB000010	PIO Command Register

5. PIO Operation

5.1 Software Control Mode

In software control mode data input/output and control signals are all controlled by software.

5.1.1 Data Input from External Device

1) Write "1x" to the PCMD (Command Register) PMOD bits ("x" indicates that either "0" or "1" is acceptable).

2) Read the input buffer PIB to read the parallel port's pin levels at that time.

5.1.2 Data Output to External Device

1) Write "1x" to the PCMD (Command Register) PMOD bits ("x" indicates that either "0" or "1" is acceptable).

2) Write a value to the output buffer POB.

3) Write "1" to the bits in PDIR (Direction Register) that correspond to parallel port pins that will be outputs. This starts to drive the parallel port for data to be output.

4) If "0" is written to any bits in PDIR, then the corresponding parallel port pins will stop being driven.

5.2 Handshake Control Mode

In handshake control mode data input is controlled by handshake using a strobe ($\overline{\text{PSTB}}$), input buffer full (PIBF), acknowledge ($\overline{\text{PACK}}$), and output buffer full (POBF) for input/output.

5.2.1 Data Input from External Device

(A) SCP operation

1) Write "01" to the PCMD (Command Register) PMOD bits. Also write "1" to PCMD's PIEN bit to enable input operation.

2) When data is written to the input buffer PIB from the external device, the PSTS (Status Register) PIST bit will become "1" to indicate that there is valid data in PIB. When PSTS's PIST bit becomes "1", the input buffer full output (PIBF) will become "1".

3) If PSTS's PIST bit is "1" and input buffer full interrupts have been enabled (PCMD's PIIE bit is "1"), then a PIO interrupt to the CPU core will be generated.

4) The CPU core verifies that PSTS's PIST bit is "1" in the PIO interrupt vector process routine and reads the input buffer PIB. It then writes "0" to PSTS's PIST bit to release the interrupt.

5) When PSTS's PIST bit becomes "0", the input buffer full output (PIBF) also becomes "0".

6) Repeat the operation from step 2).

(B) External operation

1) Verify that the input buffer full output (PIBF) is "0".

2) Drive the parallel input/output bus (PD[7:0]) with input data.

3) Set the strobe input ($\overline{\text{PSTB}}$) to "1". This writes the data to the input buffer PIB.

4) When the input buffer full output (PIBF) becomes "1", stop driving the parallel input/output bus and set the strobe input ($\overline{\text{PSTB}}$) to "0".

5) Repeat the operation from step 1).

5.2.2 Data Output to External Device

(A) SCP operation

1) Write "01" to the PCMD (Command Register) PMOD bits. Also write "1" to PCMD's POEN bit to enable output operation.

2) Verify that the PSTS (Status Register) POST bit is "1", indicating that the output buffer POB is empty.

3) Write data to the output buffer POB.

4) Write "0" to PSTS's POST bit. When POST becomes "0", the output buffer full output (POBF) will become "1".

5) If PSTS's POST bit is "1" and output buffer full interrupts have been enabled (PCMD's POIE bit is "1"), then a PIO interrupt to the CPU core will be generated. This interrupt will be released by writing "0" to PSTS's POST bit or writing "0" to PCMD's POIE bit.

6) Write data to the output buffer POB and repeat the operation from step 4).

(B) External operation

1) Verify that the output buffer full output (POBF) is "1".

2) Set the acknowledge input ($\overline{PACK}$) to "1". When the acknowledge input is set to "1", the PIO will output the value of the output buffer (POB) to the parallel input/output bus (PD[7:0]).

3) Verify that the output buffer full output (POBF) is "0".

4) Read the value on the input/output bus.

5) Set the acknowledge input ($\overline{PACK}$) to "0". When the acknowledge input becomes "0", the PIO will stop driving the input/output bus.

6) Repeat the operation from step 1).

5.3 Bus Control Mode

In bus control mode data input/output is controlled externally by the chip select input ($\overline{PCS}$), flag/buffer select input (PIOA), read input ($\overline{PACK}$), and write input ($\overline{PSTB}$).

(A) SCP operation

1) Write "00" to the PCMD (Command Register) PMOD bits. Also write "1" to PCMD's PIEN bit to enable input operation.

2) When an external device writes data to the input buffer PIB, the PSTS (Status Register) PIST bit will become "1", indicating that there is valid data in the input buffer PIB.

3) If PCMD's PIIE bit is "1", then a PIO interrupt to the CPU core will be generated.

4) The CPU core verifies that PSTS's PIST bit is "1" in the PIO interrupt vector process routine and reads the input buffer PIB. It then writes "0" to PSTS's PIST bit to release the interrupt.

5) Repeat the operation from step 2).

(B) External operation

1) Read the input buffer full output (PIBF).

Chip select input	$\overline{PCS}$	0
Flag/buffer select input	PIOA	0
Read input	$\overline{PACK}$	0
Write input	$\overline{PSTB}$	1

2) Verify that the input buffer full output (PIBF) is "0".

3) Write data to the input buffer (PIB).

Chip select input	$\overline{PCS}$	0
Flag/buffer select input	PIOA	1
Read input	$\overline{PACK}$	1
Write input	$\overline{PSTB}$	0
Input/output bus	PD[7:0]	Write data

4) Repeat the operation from step 1).

5.3.2 Data Output to External Device

(A) SCP operation

1) Write "00" to the PCMD (Command Register) PMOD bits. Also write "1" to PCMD's POEN bit to enable output operation.

2) Verify that the PSTS (Status Register) POST bit is "1", indicating that the output buffer POB is empty.

3) Write data to the output buffer POB.

4) Write "0" to PSTS's POST bit. When POST is "0", the output buffer full output (POBF) will become "1".

5) When POST becomes "1" and PCMD's POIE bit is "1", then a PIO interrupt to the CPU core will be generated. This interrupt will be released by writing "0" to PSTS's POST bit or by writing "0" to PCMD's POIE bit.

6) Repeat the operation from step 3).

(B) External operation

1) Read the output buffer full output (POBF).

Chip select input	$\overline{PCS}$	0
Flag/buffer select input	PIOA	0
Read input	$\overline{PACK}$	0
Write input	$\overline{PSTB}$	1

2) Verify that the output buffer full output (POBF) is "1".

3) Read the output buffer (POB).

Chip select input	$\overline{PCS}$	0
Flag/buffer select input	PIOA	1
Read input	$\overline{PACK}$	0
Write input	$\overline{PSTB}$	1
Input/output bus	PD[7:0]	Read data

4) When the output buffer is read, PSTS's POST bit will become "1".

5) Repeat the operation from step 1).

Timer Unit

1. Features

The timer unit (TMR) is a 16-bit programmable timer. It has two modes: an interval timer mode which requests interrupts to the CPU core, and a clock division mode which generates a 50% duty, frequency divided clock.

2. TMR Functions

2.1 Counter

- 16-bit up counter

2.2 Counter Clock Period

- Φ (SCP operating frequency × 1)
- 4Φ (SCP operating frequency × 4)
- 16Φ (SCP operating frequency × 16)
- 64Φ (SCP operating frequency × 64)

2.3 Interval Timer Interrupts

When the counter overflows (counter value changes from 0xFFFF to 0x0000), it will request an interrupt to the CPU core.

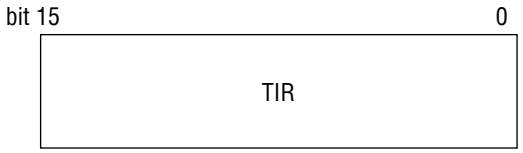
2.4 Divided Clock Generation

When the counter overflows (counter value changes from 0xFFFF to 0x0000), it will invert the value currently being output.

3. TMR Registers

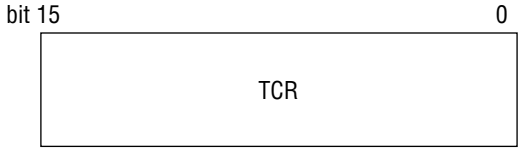
3.1 TIR: Timer Initial Value Register

This register saves the counter's initial value. When this register is written, the same value will be written to the Timer Value Register (TCR). This register will be undefined after reset.



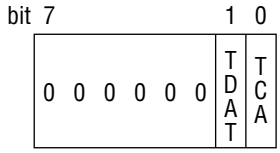
3.2 TCR: Timer Value Register

This register provides the counter's current value. It will be undefined after reset.



3.3 TSTS: Timer Status Register

This register provides TMR status.



• bit[1] TDAT

0 : No timer interrupt request occurred (interval timer mode)
Divided clock output is "0" (divided clock mode)

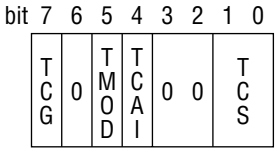
1 : Timer interrupt request occurred (interval timer mode)
Divided clock output is "1" (divided clock mode)

• bit[0] TCA

0 : Timer counter operation is suspended
1 : Timer counter is operating

3.4 TCMR: Command Register

This register sets TMR operation.



• bit[7] TCG

0 : Disable timer counter operation

1 : Enable timer counter operation

This bit will be "0" after reset.

• bit[5] TMOD

0 : Interval timer mode
1 : Divided clock mode

This bit will be "0" after reset.

• bit[4] TCAI

0 : Disable auto-initialization of timer value register
1 : Enable auto-initialization of timer value register

This bit will be "0" after reset.

• bit[1:0] TCS

00 : Count clock Φ
01 : Count clock 4 Φ
10 : Count clock 16 Φ
11 : Count clock 64 Φ

These bits will be "00" after reset.

4. TMR Register Addresses

The MSM7630 has two timers. The register addresses for each are listed below.

TMR1	Timer Initial Value Register	0xF8000000
	Timer Value Register	0xF8000004
	Timer Status Register	0xF8000008
	Timer Command Register	0xF800000C
TMR2	Timer Initial Value Register	0xF8000010
	Timer Value Register	0xF8000014
	Timer Status Register	0xF8000018
	Timer Command Register	0xF800001C

5. TMR Operation

5.1 Interval Timer Mode

In interval timer mode counting begins from the value set in the Timer Initial Value Register, and a timer interrupt is generated to the CPU when the counter overflows.

1) Set the TCMR (Command Register) TCG bit to "0", disabling counting.

2) Set TCMR's TCS bits to select the counter's increment clock.

3) Set TCMR's TMOD bit to "0", setting interval timer mode as the operating mode.

4) To generate periodic interrupts set TCMR's TCAI bit to "1", which will set the counter to be loaded with the value of the Timer Initial Value Register each time the counter overflows. For a one-shot interrupt set the TCAI bit to "0".

5) Set the timer's initial value in the Timer Initial Value Register TIR. Writing to this register will simultaneously write the same value to the Timer Value Register TCR.

6) Write "1" to TCMR's TCG bit to start counting. An interrupt will be generated when the counter overflows.

7) To release the interrupt set the TSTS (Status Register) TDAT bit to "0" in software.

5.2 Divided Clock Mode

In divided clock mode counting begins from the value set in the Timer Initial Value Register, and the divided clock output value inverts when the counter overflows.

1) Set the TCMR (Command Register) TCG bit to "0", disabling counting.

2) Set TCMR's TCS bits to select the counter's increment clock.

3) Set TCMR's TMOD bit to "1", setting divided clock mode as the operating mode.

4) To generate a periodic divided clock set TCMR's TCAI bit to "1", which will set the counter to be loaded with the value of the Timer Initial Value Register each time the counter overflows. For a one-shot divided clock set the TCAI bit to "0".

5) Set the timer's initial value in the Timer Initial Value Register TIR. Writing to this register will simultaneously write the same value to the Timer Value Register TCR.

6) Write "1" to TCMR's TCG bit to start counting and generating a divided clock.

Speech Data Registers

1. Features

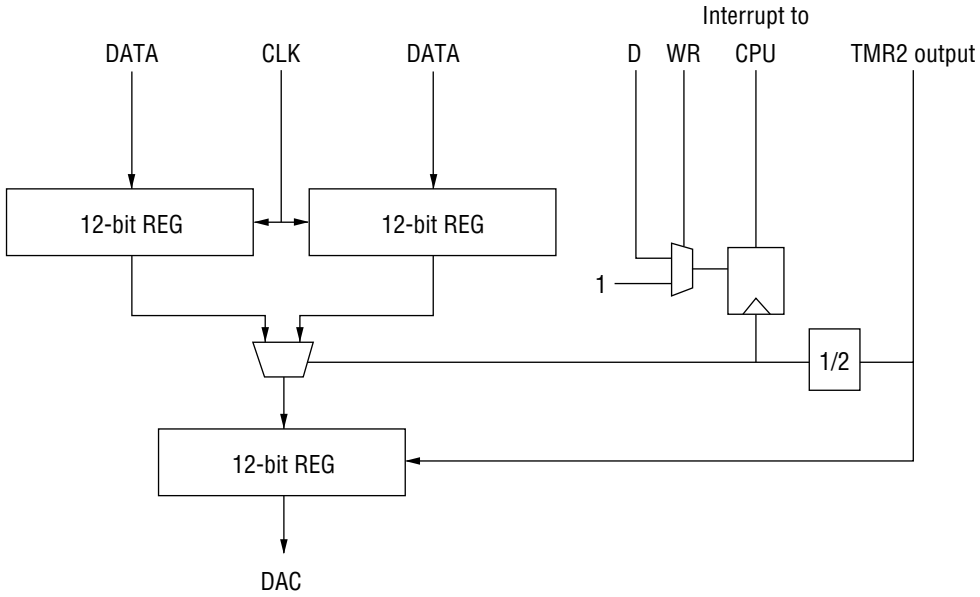
This is a register group and control circuit used for speech output.

2. Speech Data Registers Functions

2.1 Speech Output Registers

These are 12-bit registers that store speech output data. There are two registers and one output register configured to operate at the speech sampling frequency.

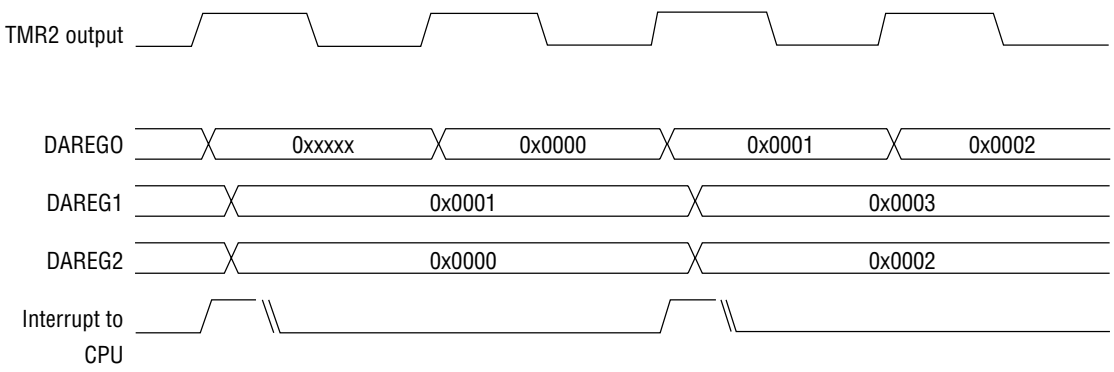
Use of two registers reduces the frequency of interrupt generation during waveform output, which lightens the CPU load. The output stage is provided in the register, which corrects the inaccuracies in the sampling frequencies that are caused by interrupts.



The two registers are in parallel, continuously written with D/A conversion data. The output register reads the data from the two registers alternately in every sampling cycle.

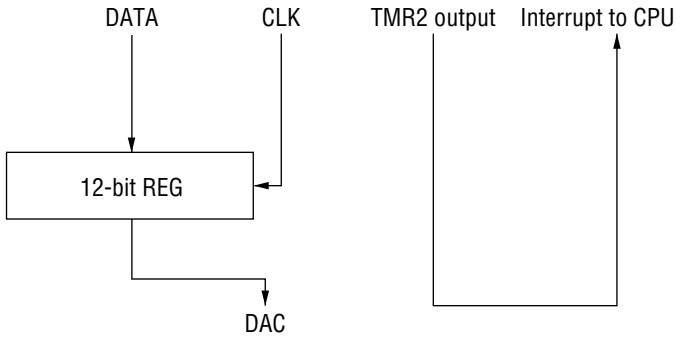
The output level registers' clock is generated by TMR2. This clock is multiplied by 1/2 to output interrupt signals. The interrupt signals are used to write the waveform output data to the speech output registers.

Note the following when the MSM7576 mode (described later) is not used when using the speech output registers :
– The TMR2 must be set to the divided clock mode.
– To write data to DAC1 and DAC2, write to DAC2 first, then DAC1.
– Do not clear the status register of the TMR2. If it is cleared by an interrupt routine, the sampling frequency for the speech output will change.



2.2 MSM7576 Mode

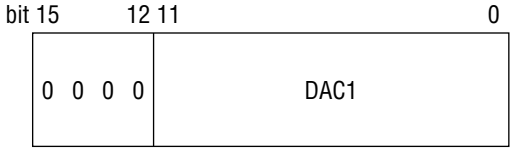
This mode forces operation to be the same as MSM7576 operation. Interrupt signals from TMR2 are output directly as interrupts to the CPU.



3. Speech Data Registers Details

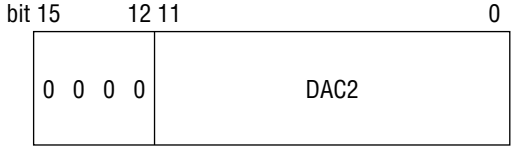
3.1 DAC1: Speech Output Register 1

This register stores speech output data. It will be "000000000000" after reset.



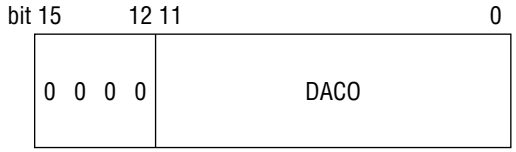
3.2 DAC2: Speech Output Register 2

This register stores speech output data. It will be "000000000000" after reset.



3.3 DACO: D/A Conversion Register

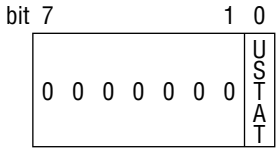
This register stores data to be input to the D/A converter. It will be "000000000000" after reset.



3.4 USTAT: Status Register

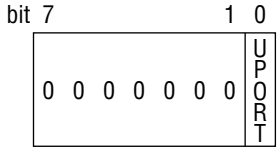
This register indicates whether or not the speech output registers/circuits have generated an interrupt to the CPU.

Writing "0" to this register releases the interrupt from the speech output registers/circuits. In MSM7576 mode USTAT will become "0" when the TMR2 interrupt is released.



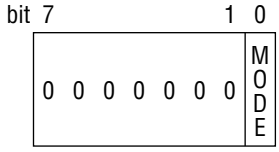
3.5 UPORT: General Register

This is a general register. It will be "0" after reset.



3.6 MODE7576: MSM7576 Mode Select Register

This register sets MSM7576 mode. It will be "0" after reset.



4. Speech Output Registers Address Configuration

The addresses used by the speech output registers/circuits are assigned at 0x80000000. Accesses to this space will be 3τ access.

Speech Output Register 1	0x80000000
Speech Output Register 2	0x80000004
D/A Conversion Register	0x80000008
Status Register	0x8000000C
General Register	0x80000010
MSM7576 Mode Select Register	0x80000020

Speech Output

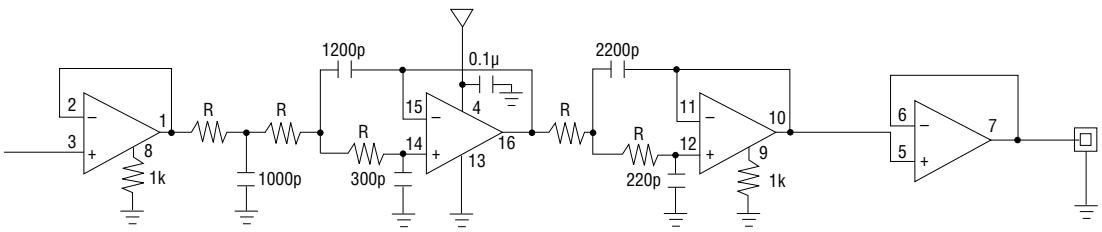
1. Output Waveform from DAO1

The speech output pin directly outputs the output of the DA converter.

The output waveform from DAO1 will be a staircase synchronized to the sampling frequency. Maximum output amplitude will be ($4095/4096 \times V_{DD}$).

2. Output Filter

Because the output from DAO1 is a staircase described above, add a low-pass filter. The diagram below shows a reference circuit for a Butterworth low-pass filter.



MC14573P

Butterworth low-pass filter

R = 47k, f = 4.8 kHz (95 model: for 12 kHz sampling)

R = 36k, f = 6.4 kHz (96 model: for 16 kHz sampling)

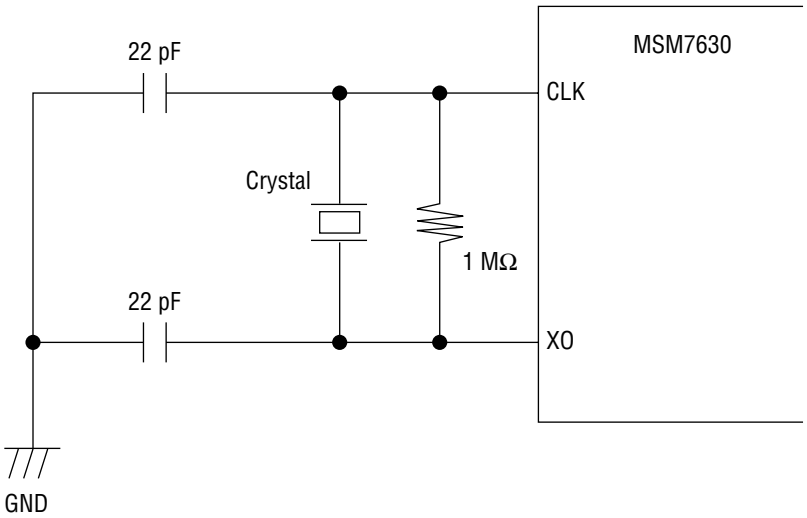
R = 27k, f = 9.6 kHz (97 model: for 22 kHz sampling)

Oscillation Circuit

There are two methods to generate the MSM7630 system clock: adding an external crystal oscillator or supplying an external clock.

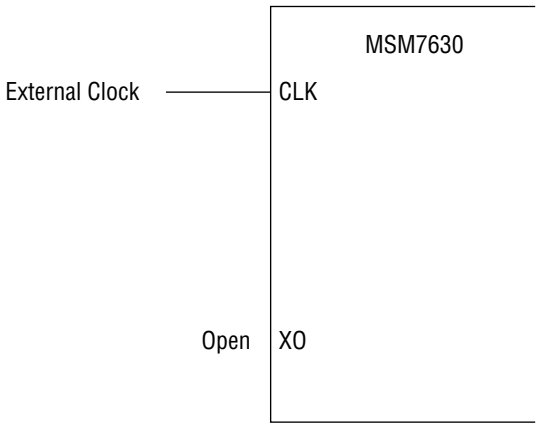
1. Crystal Oscillator

The diagram below shows a connection example for a crystal oscillator.



2. External Clock

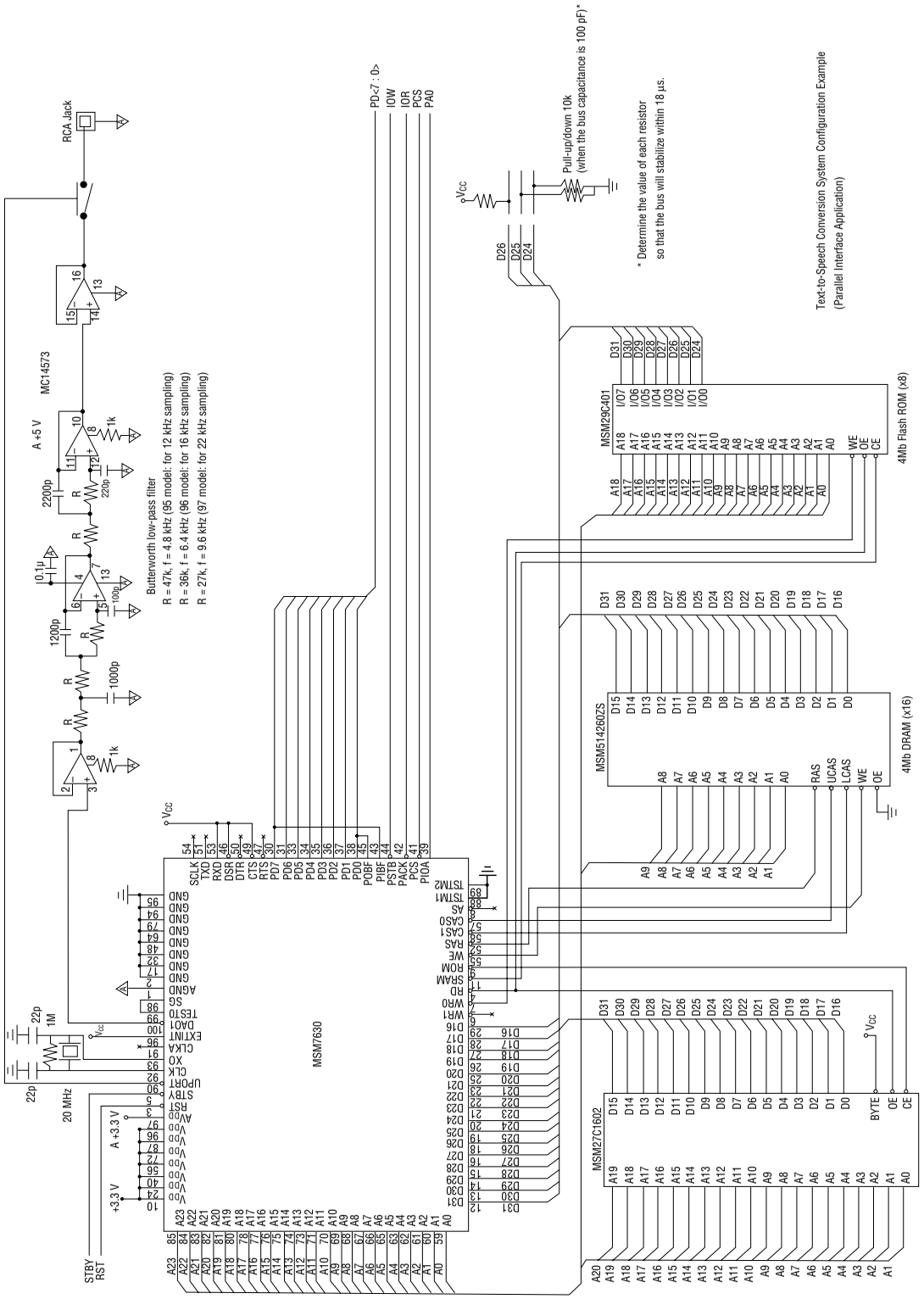
The diagram below shows an example using an external clock.



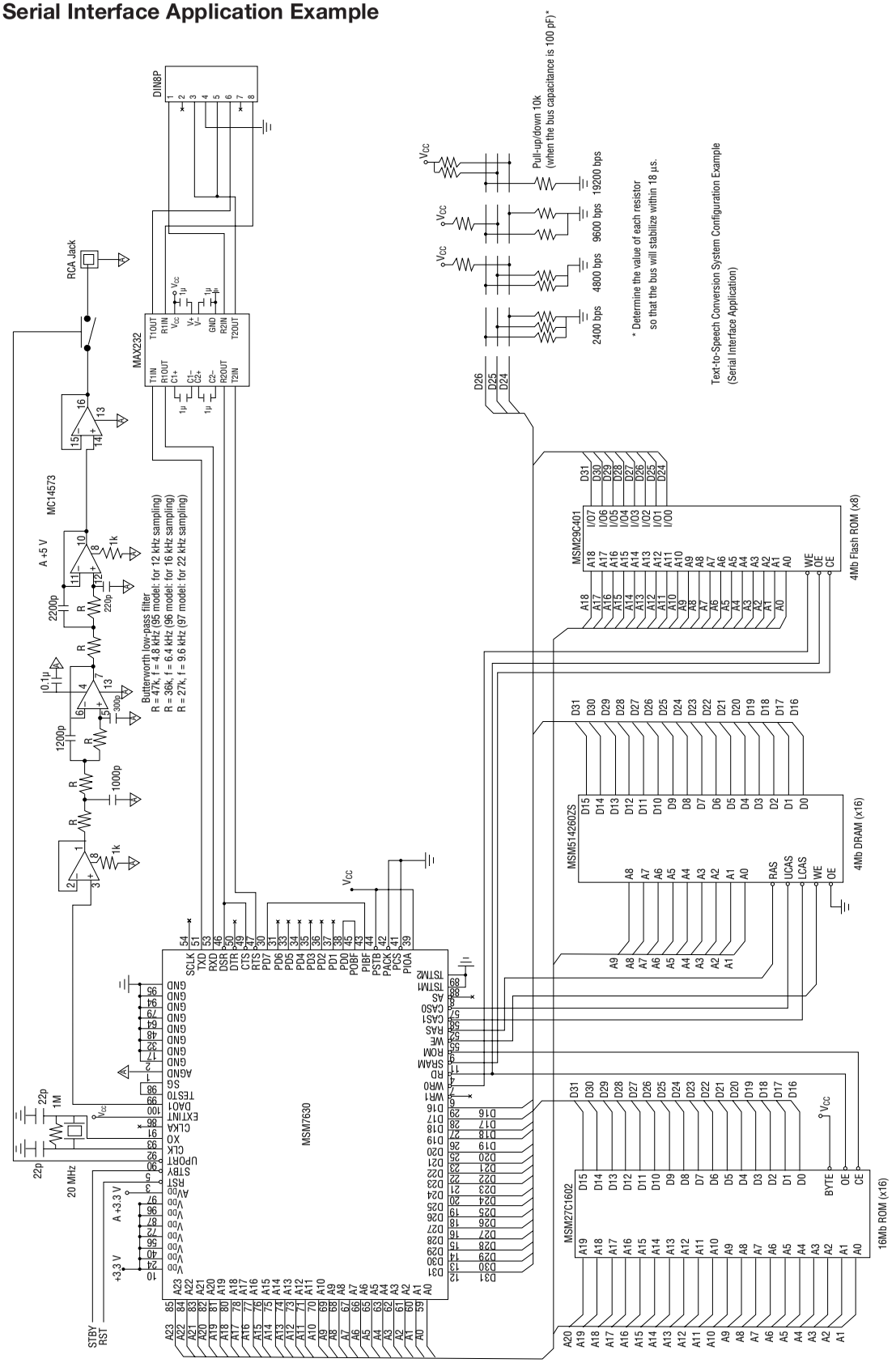
The external clock is input on the CLK pin. Leave the XO pin open.

SYSTEM CONFIGURATION EXAMPLE

Parallel Interface Application Example



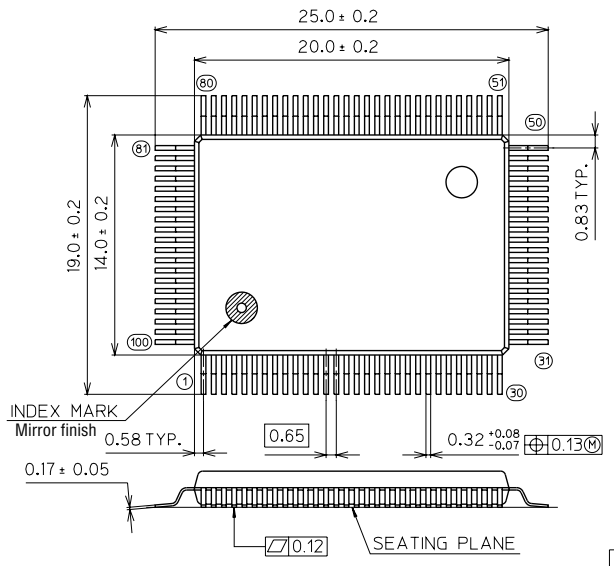
Serial Interface Application Example

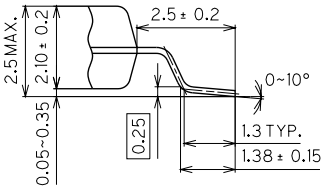


PACKAGE DIMENSIONS

(Unit : mm)

QFP100-P-1420-0.65-BK





Package material	Epoxy resin
Lead frame material	42 alloy
Pin treatment	Solder plating
Solder plate thickness	5 μm or more
Package weight (g)	1.29 TYP.

Notes for Mounting the Surface Mount Type Package

The SOP, QFP, TSOP, TQFP, LQFP, SOJ, QFJ (PLCC), SHP, and BGA are surface mount type packages, which are very susceptible to heat in reflow mounting and humidity absorbed in storage. Therefore, before you perform reflow mounting, contact Oki's responsible sales person on the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).