MX93132 DATA SHEET

CONTENT

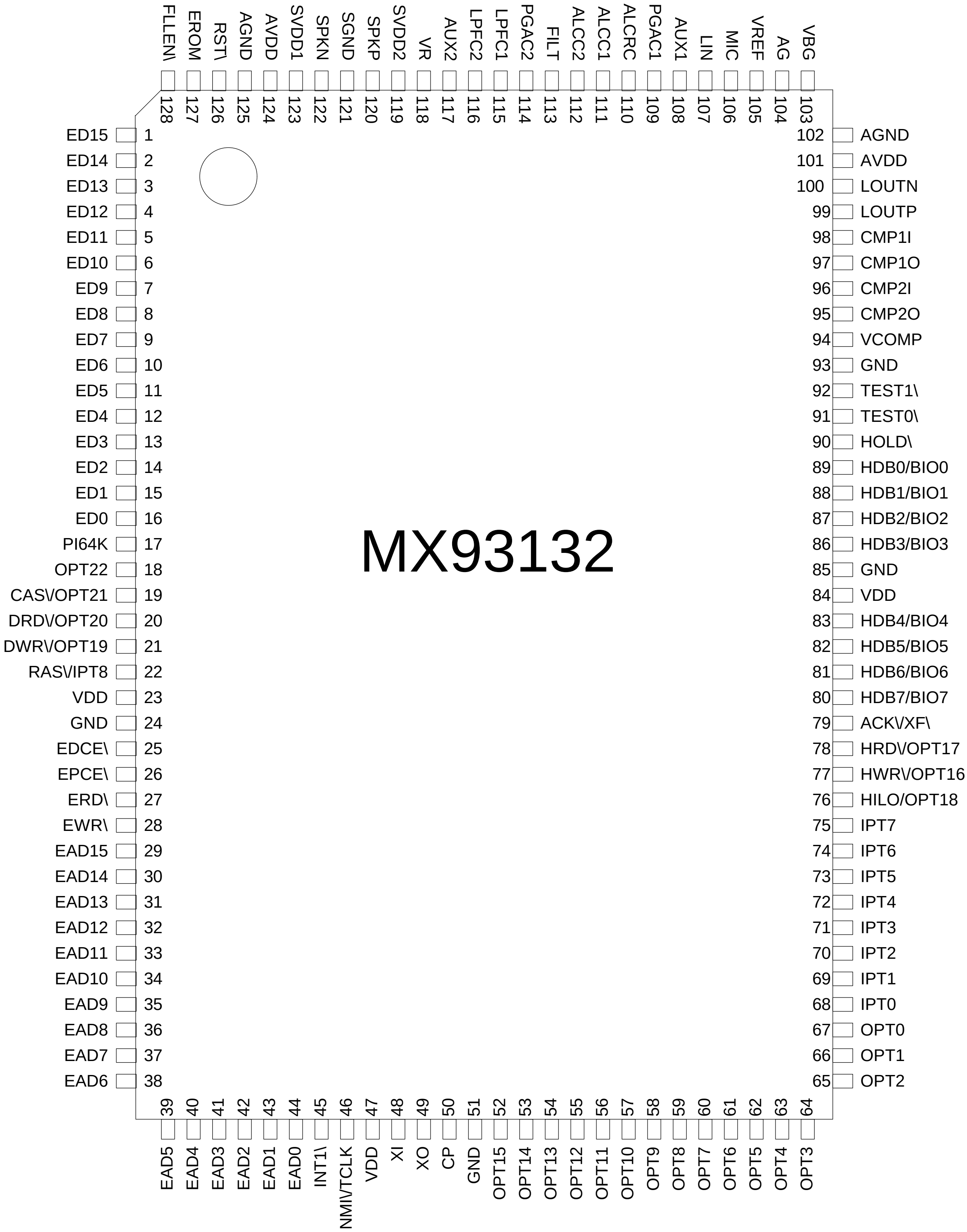
1.1 FEATURES

- Optimized for highly integrated digital answering machine application
- Built in DRAM controller; interface with x1,x4, x8 and x16 configuration
- One 8 bits host interface
- Maximum 9 general input pins , 23 output pins and 8 programmable bi-directional I/O pins
- One external interrupt pins
- 1ms internal timer interrupt
- 64 K words program space, 64K internal , in which control code and voice prompt can be built
- 64 K words data space , 2.5 K words data RAM internal
- 45 MHz running clock , provide 30 MIPs processing power with 40 mA active current
- Built in FLL with 4.096 MHz clock as clock source to achieve 2 mA consumption in power down mode operation
- 16 x 16 multiplication and 32 bit accumulation executed in one instruction cycle
- Single cycle normalization instruction
- 32 bit barrel shifter with left/right shift 15 bits capability
- 32 level hardware stack
- 8 Auxiliary registers used in register indirect addressing.
- Zero-overhead hardware looping , maximum 8 instruction words executed repeatedly 1024 times maximum
- Built-in two PCM CODECs
- Support Digital Speakerphone application
- CODECs support 16-bit format linear data
- Support switch paths for DAM (digital answering machine) related applications
- Support two comparators for power-low and battery -low detection
- Support external L..P.F. for D/A output path
- Support external volume control
- On-chip differential line driver
- On-chip ALC (automatic level control)
- On-chip digital volume control of CODEC
- On-chip programmable receive/transmit gain control of CODEC
- Easy interface to FAX or cordless Phone
- Fabricated in 0.5 um 5V CMOS process
- 128 pins PQFP package

1.2 DIFFERENCE between MX93011C and MX93132

	MX93011C	MX93132
INTERNAL RAM SIZE	2K Words Bank0 : 0x0000 ~ 0x03FF(1K) Bank1 : 0x0400 ~ 0x07FF(1K)	2.5K Words Bank0 : 0x0000 ~ 0x03FF(1K) Bank1 : 0x0400 ~ 0x09FF(1.5K)
EXTERNAL RAM STARTING ADDRESS	0X0800	0X1000
INTERNAL ROM SIZE	32k Words	64k Words
EXTERNAL ROM STARTING ADDRESS	0X8000	No external ROM
REPEAT COUNT REGISTER	7-BIT	10-BIT
AR MODULO REGISTER	7-BIT	10-BIT
INTERRUPT PENDING STATUS REGISTER	REG5 (R)	No
CONTINUOUS INSTRUCTION "SQRA"	Overflow problem	Fix continuous "SQRA"
EXTENDED OUTPUT PORT REGISTER	OPT21 – OPT19	OPT22 – OPT19
CODEC COMMAND REGISTER	No	REG5(R/W)
CODEC RECEIVE/TRANSMIT REGISTERS	REG16(R) : CDRR0 REG17(W) : CDXR0	REG16(R/W): CDDR0, CDXR0 REG17(R/W): CDDR1, CDXR1
CODEC INTERFACE	Single external codec interface	Two built-in internal codec
X' TAL source	32.256MHz & 32.768KHz	4.096MHz
FLL Multiplication Factor Register (FLLMR)	13-Bit (0 – 0x1FFF)	5-Bit (12 – 24)
FLL Control Register (FLLCONR)	12-Bit	No
FLL Status Register (FLLSR)	13-Bit	No
CMCK Divide Ratio Register (CMCKDIVR)	5-Bit	No

2.1 PIN OUT for 128 PIN PQFP MX93132



2.2 PIN DESCRIPTIONS

1. POWER/CLOCK/CONTROL PINS :			
Name	**Pin Type**	**Pin Number**	**Description**
VDD	Power	23,47,84	5 Volt power source pins
GND	Power	24,51,85,93	Ground pins
FLLEN\	IS	128	1 : Test X‘ tal mode. 0 : Single low X‘ tal mode. High clock will be generated from FLL
XI	X‘ tal	48	4.096 MHz crystal oscillator‘ s input
XO	X‘ tal	49	4.096 MHz crystal oscillator‘ s output
CP	I/O(A)	50	Output of internal PLL charge pump circuit.
RST\	IS	126	Power on reset pin.Minium timing 50ms.
HOLD\	IS	90	Level trigger.Hold down clock to DSP (X‘ tal oscillator or FLL is still active) and related data ,address and control pins will go to high-impedance state.
EROM	IS	127	Map all program memory space to external
PI64K	IS	17	Select Internal ROM size (High : 64K, Low : 48K)
NMI\/TCLK	IS	46	Falling Edge-triggered non-maskable external interrupt / Test clock in
INT1\	IS	45	Falling Edge-triggered maskable external interrupt
TEST0\	ISH	91	Test pin for CODEC
TEST1\	ISH	92	Test pin for CODEC

Note 1: FLLEN\,HOLD\,EROM,GND,NMI\/TCLK,INT1\,TEST0\,TEST1\ pin output low when DSP is in reset state or in power down mode.

2. CODEC INTERFACE PINS :			
Name	**Pin Type**	**Pin Number**	**Description**
AVDD	Power	101,124	5V power for analog circuit
SVDD1	Power	123	5V power for speaker driver
SVDD2	Power	119	5V power for speaker driver
AGND	Power	102,125	Ground for analog circuit
SGND	Power	121	Ground for speaker driver
VCOMP	I(A)	94	Reference voltage for voltage comparator
CMP2O	O(A)	95	Voltage comparator 2 output
CMP2I	I(A)	96	Non-inverting input of voltage comparator 2
CMP1O	O(A)	97	Voltage comparator 1 output
CMP1I	I(A)	98	Non-inverting input of voltage comparator 1
LOUTP	O(A)	99	Non-inverting output of **LIN-DRV** with PGA; PGA from 0 to 22.5 dB; 1.5 dB/step.
LOUTN	O(A)	100	Inverting output of LIN-DRV with PGA; PGA from 0 to 22.5 dB; 1.5 dB/step.
VBG	O(A)	103	Band-gap reference; normal 1.25V and should not be used to sink or source current
AG	O(A)	104	Internal analog signal ground; normal 2.25V and should not be used to sink or source current.
VREF	O(A)	105	Voltage reference; normal 2.25V and can sink 450uA
MIC	I(A)	106	Microphone input with **PRE-PGA**; PGA from -15 to 21 dB
LIN	I(A)	107	Telephone line signal input with **PRE-PGA**; PGA from -15 to 21 dB
AUX1	I/O(A)	108	Auxiliary signal input with **PRE-PGA**; PGA from -15 to 21 dB
PGAC1	O(A)	109	programmable gain amplifier(PRE-PGA) compensate capacitor
ALCRC	O(A)	110	Automatic level control (**ALC**) time constant
ALCC1	O(A)	111	Automatic level control (**ALC**) DC blocking capacitor output
ALCC2	O(A)	112	Automatic level control (**ALC**) DC blocking capacitor input
FILT	I/O(A)	113	1.anti-aliasing filter; 2. As an I/O port for **AIN** (A/D input)
PGAC2	O(A)	114	Programmable Gain Amplifier Offset Capacitor
LPFC1	O(A)	115	Option of external passive **L.P.F** (Low Pass Filter);
LPFC2	O(A)	116	Option of external passive **L.P.F** (Low Pass Filter);
AUX2	I/O(A)	117	I/O port for **SWK** and **SWH**
VR	O(A)	118	External speaker volume control; use a variable 10K variable resistor.
SPKP	O(A)	120	Inverting output of SPK-DRV with DA-PGA, ATT1 And ATT2; PGA from 0 to 9 dB; Attenuator 1 & 2 from 0 to –45 dB.
SPKN	O(A)	122	Non-inverting output of SPK-DRV with DA-PGA, ATT1 And ATT2; PGA from 0 to 9 dB; Attenuator 1 & 2 from 0 to -45 dB.

3. MEMORY INTERFACE PINS :			
Name	**Pin Type**	**Pin Number**	**Description**
EAD[15:0]	OA/Z	29-44	External memory address bus. Note 2
ED[15:0]	IT/OA/Z	1-16	External memory data bus. Note 2
EDCE\	OA/Z	25	External data memory chip enable. Note 2
EPCE\	OA/Z	26	External program memory chip enable. Note 2
ERD\	OA/Z	27	External memory read enable. Note 2
EWR\	OA/Z	28	External data memory write enable. Note 2
CAS\	OA	19	DRAM column address select
RAS\	OA/Z	22	DRAM row address select
DRD\	OA	20	DRAM read enable
DWR\	OA	21	DRAM write enable

Note 2: Placed in high-impedance state when DSP is in HOLD mode.

4. PARALLEL INTERFACE (HOST INTERFACE) PINS : When HOSTM bit in CTLR =0			
Name	**Pin Type**	**Pin Number**	**Description**
HDB[7:0]	IS/OA/Z	80-83,86-89	Parallel data bus to external host controller
HILO	IS/OA/Z	76	High or low byte select. 1: select high byte 0: select low byte
HRD\	IS/OA/Z	78	Host read enable
HWR\	IS/OA/Z	77	Host write enable
ACK\	OA	79	Acknowledge to external host that there is response from DSP to be read by external host.

5. GENERAL PURPOSE I/O PORT PINS			
Name	**Pin Type**	**Pin Number**	**Description**
IPT[3:0]	ISH	68-71	Input ports with internal pull high resister (R ~= 150 K ohm)
IPT[7:4]	IS	72-75	Input ports
IPT8	IS	22	Input port
OPT[15:0]	OB	52-67	Output ports
BIO[7:0]	IS/OA	80-83,86-89	Programmable bi-directional I/O ports
OPT[18:16]	OA	76-78	Output ports
OPT[21:19]	OA	19-21	Output ports
OPT22	OB	18	Output ports
XF\	OA	79	External flag. Can be changed directly by SXF/RXF instruction.

2.3 PIN TYPE ABBREVIATION :

Pin Type	Description	Pin Type	Description
IS	CMOS level schmidt trigger input buffer	OB	16 mA drive output buffer
ISH	CMOS level schmidt trigger input buffer with an internal pull high resistor built in	Z	High impedance state
OA	8 mA drive output buffer	X' tal	Crystal oscillator input/output pin
I(A)	Analog input port	O(A)	Analog output port
I/O(A)	Analog Bi-direction port		

2.4 PINS SUMMARY by PIN TYPE :

Pin Type	Signal Name	Pin Type	Description
IS	INT1\ , NMI\ , IPT[8:4], HILO, HWR\ HRD\, HOLD\, RST\, EROM, FLLEN\	OB	OPT[15:0],OPT22
ISH	IPT[3:0], TEST0\, TEST1\	IS/OA	BIO[7:0] , HDB[7:0] , OPT[18:16]
OA	CAS\, DRD\, DWR\, RAS\, ACK\	IS/OA/Z	ED[15:0]
OA/Z	EAD[15:0] , EPCE\ , EDCE\ ERD\ , EWR\ .	X' tal	XI,XO .
I(A)	VCOMP, CMP2I, CMP1I, MIC, LIN	O(A)	CMP2O, CMP1O, LOUTP, LOUTN VBG, AG, VERF, PGAC1, ALCRC ALCC1, ALCC2, PGAC2, LPFC1 LPFC2, VR, SPKN, SPKP
I/O(A)	AUX1, FILT, AUX2,CP		

2.5 MULTIPLEX PINS :

	HOSTM = 0 (uP external)		HOSTM = 1 (uP inside)	
Pin Number	**Signal Name**	**Description**	**Signal Name**	**Description**
80-83,86-89	HDB[7:0]	Host data bus	BIO[7:0]	Host data bus
76	HILO	High low byte select	OPT18	Output port
78	HRD\	Host read enable	OPT17	Output port
77	HWR\	Host write enable	OPT16	Output port
79	ACK\	Acknowledge to **HOLD**	XF\	External flag

Note : **HOSTM** is bit 1 of **CTLR ,** Its power-on reset default is **0 .**

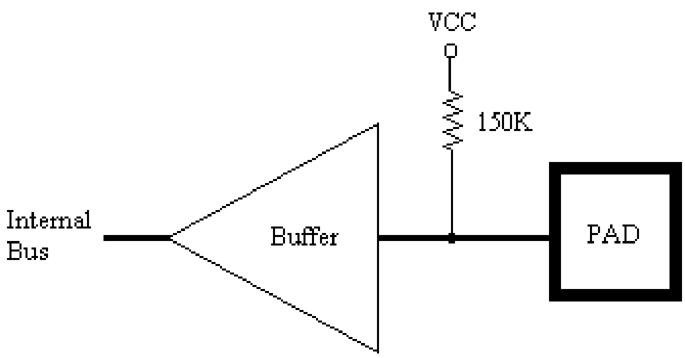
	DFS = 0 (DRAM interface)		DFS = 1 (FLASH interface)	
Pin Number	**Signal Name**	**Description**	**Signal Name**	**Description**
19	CAS\	Column address select	OPT21	Output port
20	DRD\	DRAM read enable	OPT20	Output port
21	DWR\	DRAM write enable	OPT19	Output port
22	RAS\	Row address select	IPT8	Output port

Note : **DFS** is bit 1 of **EXCTLR ,** Its power-on reset default is **0 .**

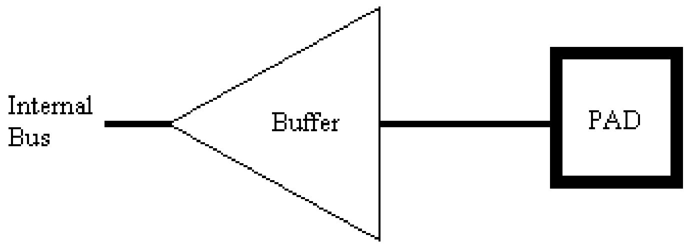
2.6 I/O PORT INTERNAL CIRCUIT :

2.6.1. Input port

Pull-high resistor : IPT0~IPT3, TEST0\, TEST1\

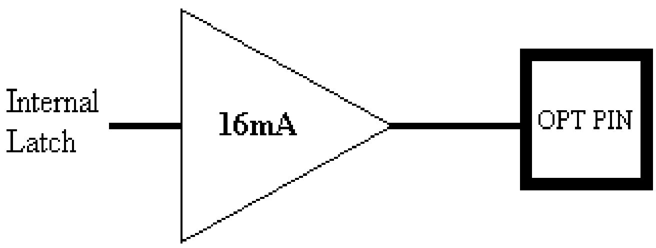


No pull-high resistor : INT1\, NMI\, IPT4~IPT7, HOLD\



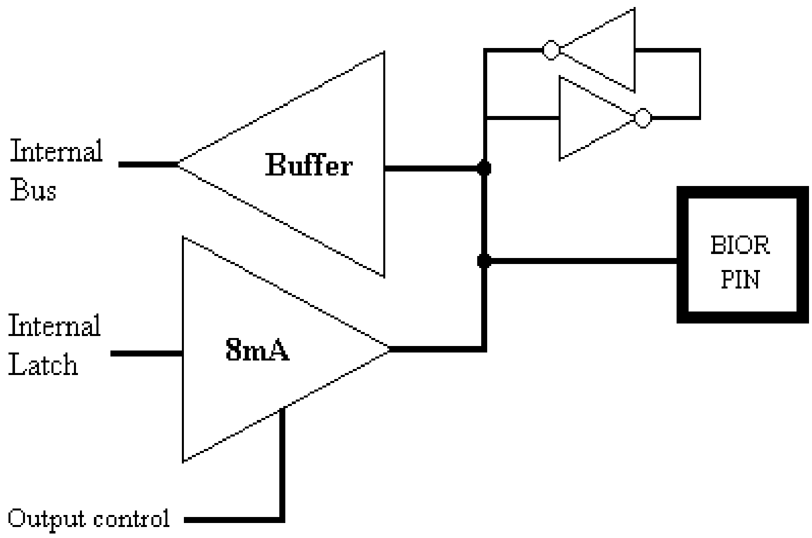
2.6.2. Output port

OPT0~OPT15



2.6.3. Bi-direction port

BIO0~BIO7,



3. ARCHITECTURE

3.1 DATA UNIT
  3.1.1 ALU
  3.1.2 ACCUMULATOR
  3.1.3 MULTIPLIER
3.2 MEMORY MAP AND ADDRESSING UNIT
  3.2.1 MEMORY MAP AND MEMORY INTERFACE
  3.2.2 IMMEDIATE ADDRESSING MODE
  3.2.3 PAGED MEMORY-DIRECT ADDRESSING
  3.2.4 REGISTER INDIRECT ADDRESSING MODE
  3.2.5 MODULO ADDRESSING
  3.2.6 MISCELLANEOUS ADDRESSING MODE
3.3 PROGRAM FLOW CONTROL UNIT
  3.3.1 CLOCK GENERATOR/FLL
  3.3.2 RUNNING MODE/PIPE LINE / WAITSTATE
  3.3.3 BRANCH/CALL/REPEAT/LOOP/STACK REGISTER
  3.3.4 INTERRUPT
    VECTOR
    MASK
    STATUS
    INTERRUPTIBLE
    NESTING
3.4 APPLICATION INTERFACE UNIT
  3.4.1 CODEC INTERFACE
  3.4.2 DRAM INTERFACE
  3.4.3 I/O FUNCTION
  3.4.4 HOST INTERFACE
  3.4.5 TIMER

3.1 DATA UNIT

3.1.1 ALU

ARITHMETIC INSTRUCTIONS:

ABS	Absolute value of high accumulator
ADH/ADHK/ADHL	Add data (from memory) or constant to high accumulator
ADL/ADLK/ADLL	Add data (from memory) or constant to low accumulator
SBH/SBHK/SBHL	Subtract data (from memory) or constant from high accumulator
SBL/SBLK/SBLL	Subtract data (from memory) or constant from low accumulator

- Execute ABS on 0x8000 will cause incorrect result, because absolute value of 0x8000 exceed the maximum positive number (0x7FFF) which can be represented.
- Data format for ALU is assumed to be signed two's complement. Short constant is treated as unsigned constant.

LOGIC INSTRUCTIONS:

OR/ORK/ORL	OR data (from memory) or constant with high accumulator
AND/ANDK/ANDL	AND data (from memory) or constant with high accumulator
XOR/XORK/XORL	Exclusive-OR data (from memory) or constant with high accumulator

DATA MOVEMENT INSTRUCTIONS:

LAC/LACK/LACL	Load data (from memory) or constant to high accumulator
SAH/SAL	Store contents of high or low accumulator to data memory
PAC	Load product register to accumulator
APAC/SPAC	Add/Subtract product register to/from accumulator
POPH/POPL	Pop top of stack to high/low accumulator
PSHH/PSHL	Push high/low accumulator onto stack

3.1.2 ACCUMULATOR

SCALING INSTRUCTIONS :

SFL/SFR/SFRS	Shift contents of accumulator left/right/right with sign extended

OVERFLOW MODE SETTING :

SOVM/ROVM	Set/Reset overflow mode

- When **OVM** bit being set , overflow mode protection is enabled. IF the results of data operation during add/subtract and shifting instructions execution exceed the maximum or minimum value that can be represented by the accumulator ,we call this condition as overflow. If overflow mode is enable in this case , data in accumulator will be saturated to the largest positive or the negative smallest number that can be represented.(0x7FFF FFFF or 0x8000 0000)

NORMALIZE INSTRUCTIONS :

NOM	Normalize contents of accumulator

- This **NOM** instruction performs hardware normalization operation on signed two's complement numbers stored in the accumulator. The left shifted counts during normalization are stored in shift count register (**SHFCR**) . **Note :** SHFCR is 5 bit wide in this normalize case , the following scaling operation by "**SFL** 0" has up to 31 bit left shift capability

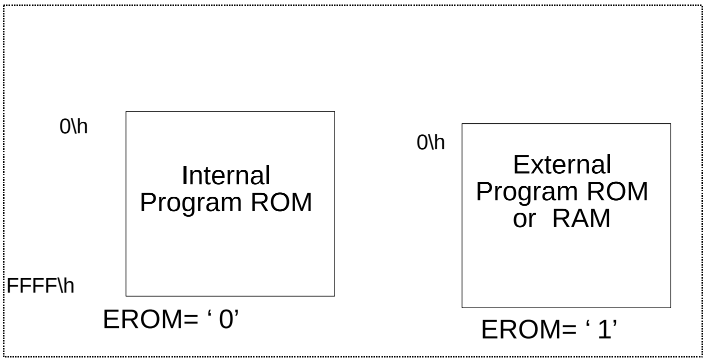
FLAG:

SIGN	MSB of high accumulator.
OV	Overflow flag for last ACCH operation. This flag will be cleared by any instructions which will generate result in accumulator.
ACZ	Accumulator zero flag. This bit reflects current accumulator status.

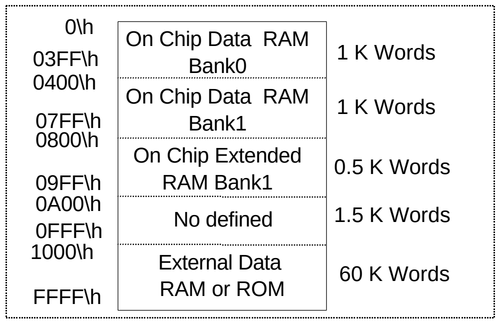
These flags are all stored in status register, and can be read out by **SSS** instruction.

3.2 MEMORY MAP AND ADDRESSING MODES

3.2.1 MEMORY MAP



PROGRAM MEMORY MAP



DATA MEMORY MAP

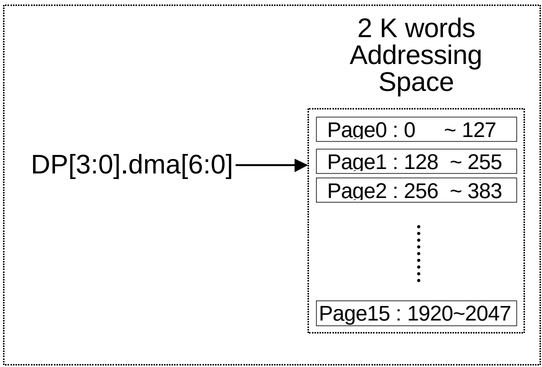
- Program memory map is selected by **EROM** pin. When EROM=1 , all program memory space are mapped to external. When EROM=0, the 64K words program memory space are totally mapped to internal **contact-programming** ROM and external program memory space does not exist.
- Program addresses 0x0000 ~ 0x000B are reserved for interrupt vector, main program can start from 0x000C.
- Totally 2.5K words internal RAM. Only first 2 K words can be accessed by short direct mode addressing. Refer to next section to see the details about data access.

3.2.2 IMMEDIATE ADDRESSING MODE

- In immediate addressing ,the immediate operand is contained in the instruction words. This immediate operand is either a un-signed 7 bit short constant or a long 16 bit constant which may be un-signed or signed(ADLL and SBLL instructions).

Example : Short immediate Long immediate
ADHK 23 ADHL 0x1234
Add 23 or 0x1234 to high accumulator

3.2.3 PAGED MEMORY DIRECT ADDRESSING



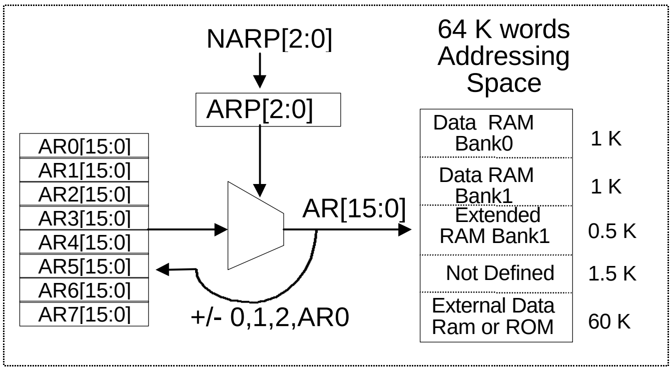
PAGED MEMORY DIRECT ADDRESSING

- In paged memory-direct addressing mode , data operand to be processed with is pointed by 11 bit address , which are composed of 4 bit data page pointer and 7 bit within-page address.
- 4 bit data page pointer **DP[3:0]** (part of **status** register) will select one of 16 pages of internal data RAM (only first two k words of internal RAM). 7 bit direct memory address is encoded in instruction word and will choose one of 128 memory location within the selected page.
- **LDP** or **LDPK** instruction can be used to modify data page pointer, **SDP** and **SSS** instructions can be used to save data page pointer in data memory.

Example : ADH 127 (if DP[3:0]=2)

Add data from memory (page 2, address within page is 127) to high accumulator

3.2.4 REGISTER INDIRECT



REGISTER-INDIRECT ADDRESSING MODE

- There are 8 auxiliary registers which are used as data memory pointer in register-indirect mode addressing. **ARP[2:0]** in status register will choose one of them as **current ar** , and this 16 bit-wide **current ar** will point to one of 64 k words data memory space in related instruction operation.
- A dedicated arithmetic unit is used to post modify the content of **current ar** parallel with instruction execution without introducing any extra instruction cycle**.** Up to seven kinds of post-modification can be made depending on what kind of operand specified in instruction word.
- **ARP[2:0]** also can be modified at the same time with new **ARP** for next following instructions use.

Syntax : **INST * [,narp]** ; Details about operand " ***** " and " **[,narp]** " are described below

Operand *	Operation
+0	No operation
-AR0	(arp) -ar0 → (arp)
+AR0	(arp)+ar0→ (arp)
+	(arp) + 1 → (arp)
-	(arp) - 1 → (arp)
++	(arp) + 2 → (arp)
--	(arp) - 2 → (arp)

Operand [,narp]	Operation
None	None
,narp	narp → arp

Note : "**[,narp]**" is an optional operand.

(arp) is one of 8 auxiliary registers which is pointed by **arp.**

Before instruction : ARP[2:0] = 5 AR5[15:0]=0x1234

Example : ADH +, 2 ; Add data from memory pointed by AR5[15:0] to high accumulator and increase AR5[15:0] by one as specified in operand " **+** " , ARP[2:0] are also updated with value "2" for following use.

After instruction : ARP[2:0] = 2 AR5[15:0]=0x1235 Note: AR2[15:0] now becomes **current ar** .

- **MAR** instruction can execute auxiliary register operation stated above alone .
- **LAR , LARK** and **LARL** instructions will load the content of specified auxiliary register with the data from memory(addressed by short direct mode or register indirect mode) or immediate constant.
- **SAR** instruction will store the content of auxiliary register specially specified to data memory(pointed by short direct mode or register indirect mode).

Special syntax : LAR *, arps [,arp]
IN * , port_address [,narp]
OUT * , port_address [,narp]

3.2.5 REGISTER INDIRECT ADDRESSING WITH MODULO ADDRESS ARITHMETIC

- Writing a non-zero value to **MODULO** register(I/O mapped 13) will enable modulo arithmetic operation in register-indirect addressing mode. A circular buffer whose length is **MOD[9:0]**+1 will be formed. This buffer starts from M-word boundaries(N*M , N=0,1,2 ...64K/K-1),where M is the smallest power of two that is equal to or greater than the size of circular buffer, and ends at buffer size location relative to start point.
- In register-indirect addressing mode operation, whenever the current auxiliary register points to the boundary of this circular buffer(either start or end boundary),it will be wrapped to the other side of the boundary for next address.
- This circular buffer must be formed in continuous memory space, that is only +/- by one post ar operation is allowed.

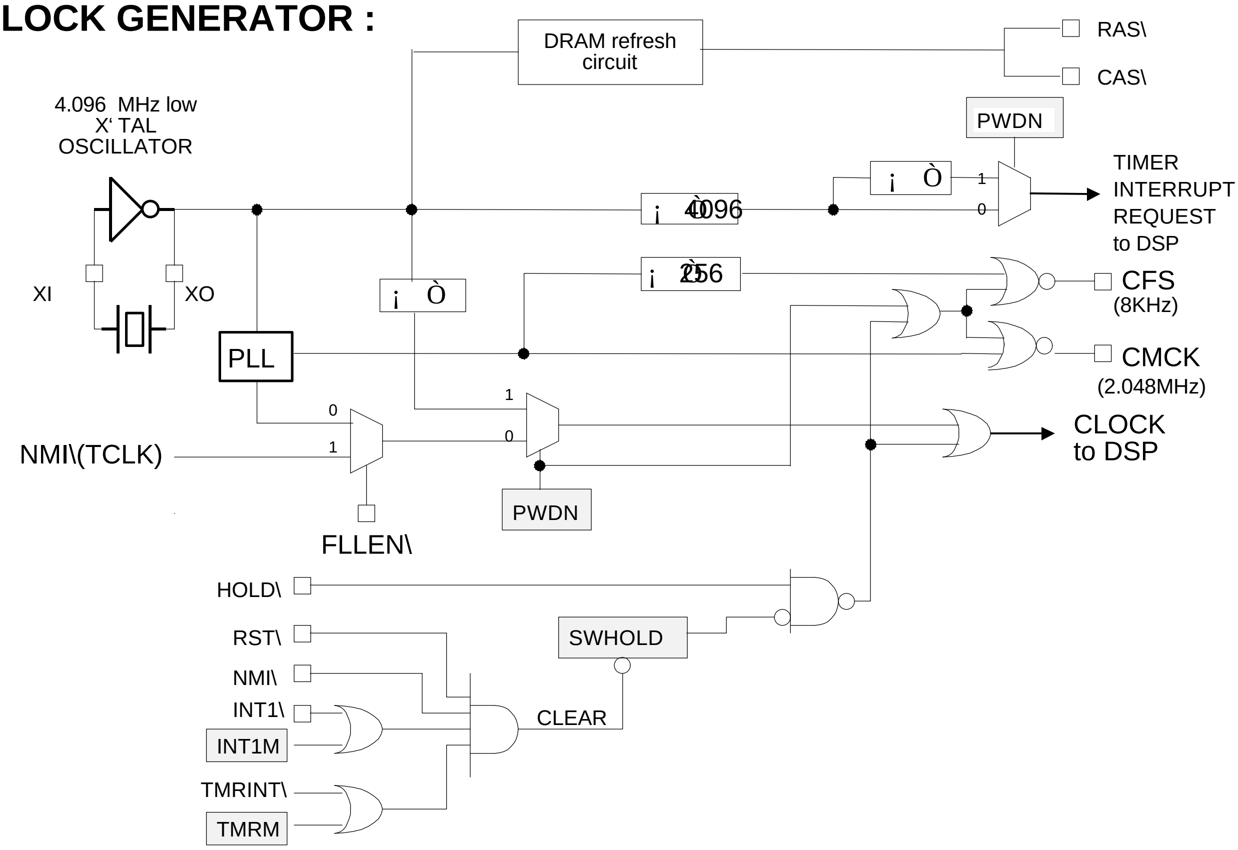
Before instruction : ARP[2:0] = 5 AR5[15:0]=0x1239 MOD[9:0]=25=0x19 M=32=0x20
AR5 just lies on the ending boundary(0x1220 ~ 0x1239)
Example : ADH +, ; Add data from memory pointed by AR5[15:0] to high accumulator
After instruction : AR5[15:0]=0x1220 (wrapped to starting boundary)

3.2.6 MISCELLANEOUS ADDRESSING MODE

- In **MB** ,**MBA** ,**MBS** multiplication instructions , the LSB of data address is decided by "**R**" or "**I**" operands. "**R**" points to the even address location, "**I**" points to the odd address location.
- In **MPA** array multiplication instructions, address of bank0 comes from **current ar** , and address of bank1(offset address) comes from **program counter** which was originally stored in high accumulator before instruction execution.

3.3 PROGRAM FLOW CONTROL UNIT
3.3.1 CLOCK GENERATOR / FLL

CLOCK GENERATOR :

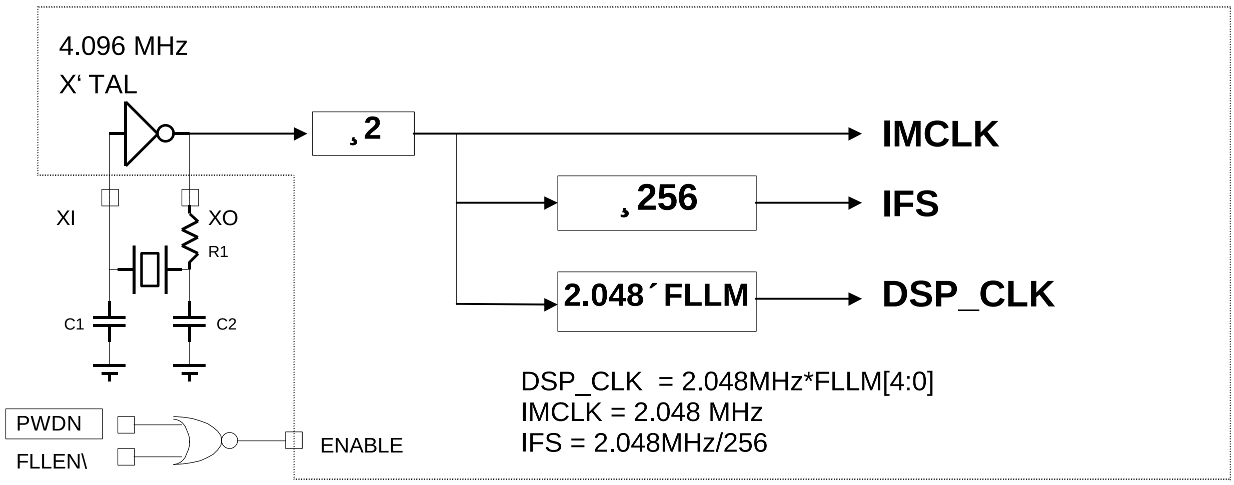


CLOCK GENERATOR FUNCTION BLOCK DIAGRAM

- In **normal** mode, clock of DSP is selected(by **FLLEN** pin =0) directly from low x' tal scillator.
  In **test clock** mode, clock of DSP is selected(by **FLLEN** pin = 1) from NMI\ pin which external test clock input.

- In **power down** mode ,clock of DSP is selected (by **PWDN** bit =1)direct from low X' tal(divided by 256). FLL will be turn off to save the power.
- In hardware or software **hold** mode (issued by **HOLD** pin or **SHOLD** bit in **CTLR**) , clock to DSP will be held down till hardware **hold** being deasserted by **HOLD** or **SHOLD** bit cleared by interrupt request. Hold mode does not save more power like power down mode does, because FLL or High X' tal is not turn off, but it responds faster for DSP resumes normal running. Timer is also active in hold mode.
- EAD[15:0],ED[15:0],EDCE\,EPCE\,ERD\ and EWR\ pins will be placed in high-impedance state when DSP is in **hold** mode.
- Details about codec clocks and timer interrupt ,please refer to section 3.4.1 and 3.4.5 .

FLL :



FREQUENCY LOCKED LOOP FUNCTION BLOCK DIAGRAM

- **FLL ENABLE :** FLL block is enabled by pin **FLLEN** =0 and will be disabled when **DSP** is in power down mode.

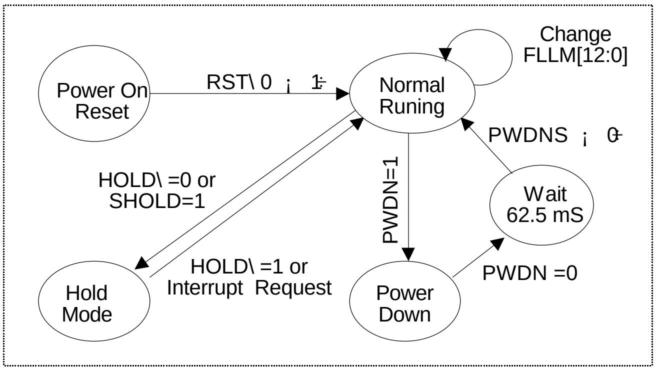
- **PROGRAMMABLE DIVIDER :** Clock from 4.096MHz X' TAL will be divided by 2 before being fed into programmable divider. Programming FLLM[4:0] register (I/O mapped 21) will change the frequency of clock to DSP based on the following equation :

  DSP_CLOCK = 4.096 MHz / 2 * FLLM[4:0]

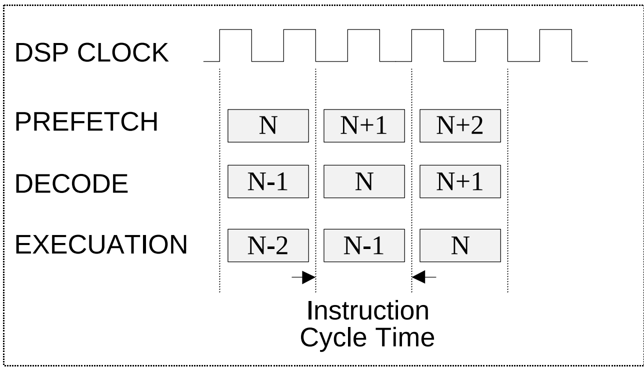
  Default : DSP_CLOCK = 4.096 MHz / 2 * 20 = 40.96 MHz

- **LOCK IN TIME :** Whenever a new frequency specified in **FLLM** register or DSP just comes back from power down mode or just starts from power on reset ,the closed loop of FLL takes about 10 mili second to lock at the target frequency.

3.3.2 RUNNING MODE/PIPE LINE/WAITSTATE



DSP RUNNING MODE



PIPE LINE and INSTRUCTION CYCLE TIME

RUNNING MODE :

- When DSP starts running from power on reset state , or change FLLM[4:0] during normal running mode , it takes about 10 ms for PLL output clock to reach the target frequency.
- When DSP wakes up from power down mode by clearing **PWDN** bit , there will be 62.5 ms lead time for DSP to switch running clock from low speed to high speed. **PWDNS** bit in **CTLR** reflects this running speed status.
- When DSP runs into hold mode either by hardware **HOLD** pin asserted low or by setting **SHOLD** bit in **CTLR** high , clock to DSP will be hold down until **HOLD** pin asserted high again or **SHOLD** bit being cleared by external interrupt or internal timer interrupt request.

PIPE LINE /WAITSTATE:

- A complete operation of instruction execution is composed of there part :
  - PREFETCH : Fetch instruction code from program ROM (either internal or external)
  - DECODE : Decode instruction and fetch data operand or store data in some location if needed
  - EXECUTION : Execute data operation in data unit.
- There are three instructions executed in parallel, each one stays in different pipeline stage. Instruction cycle is only 1/3 the time that one instruction execution really need.
- Instruction cycle time equals to the interval of one and half DSP clock for zero wait state case. Unit increase in waitstate number(for **PROGWAIT** and **DATAWAIT**), increase the instruction cycle time by one DSP clock.

3.3.3 BRANCH/CALL/REPEAT/LOOP/STACK REGISTER

BRANCH :

- **BS/BZ instructions:** Branch immediate if bit being tested equals one or zero
  Example : BS cnst3 , pma16 ; "cnst3" will be used to select one of upper byte of status register and test if condition is true or not."pma16" is new program address which DSP will jump to if condition is true.
- **BACC instruction:** Unconditional branch. After executing this instruction DSP will jump to address location specified in high accumulator.

CALL :

- **CALL instruction:** Call subroutine directly . Example : CALL pma16
- **CALA instruction:** Call subroutine indirectly. Subroutine address is specified in high accumulator.
- **Nesting CALL** is permissible and has no limit before stack overflow occurs.

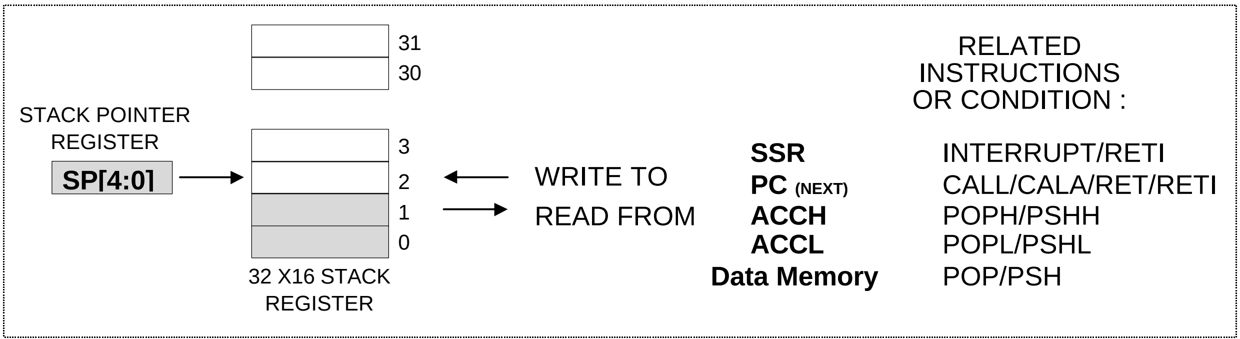
REPEAT :

- **RC :** Repeat counter. Instructions **TBR** , **MPA** and **SQRA** and instructions within program loop will be executed RC[9:0]+1 times. This repeat counter can be read by **IN** instruction and written by instructions **RPT**(RC[9:0])/**RPTK**(RC[6:0]).

LOOP :

- **LUP/LUPK instructions :** Enable hardware looping operation, and the following words (maximum 8 words) instruction will be executed RC[9:0]+1 times.
- Branch and call instructions are not allowed within program loop.

STACK REGISTER :



- Stack register size : 32x16
- 5 bit stack pointer always points to the location within stack register where next data will be put.
- Nesting call can be formed by the help of stack register to store the return address.
- No pointer overflow or underflow protection built in, when such cases occur, the pointer will be wrapped to the other side of the stack

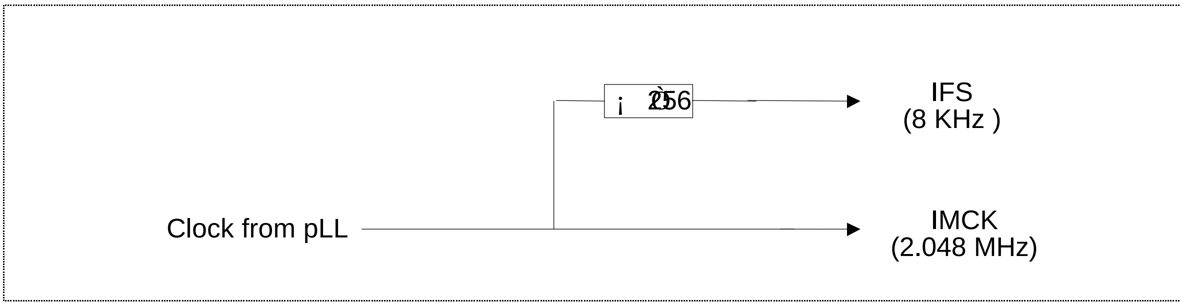
3.3.4 INTERRUPT

Interrupt Source	Vector Address	Priority	Maskable	Pending Status	Descriptions
RST\	0x0000	1st	No		Power-on reset or reset
NMI\	0x0002	2nd	No		Non-maskable interrupt
SS	0x0004	3th	Yes		Single step interrupt
INT1\	0x0006	4th	Yes	Yes	External maskable interrupt
CODECINT	0x0008	5th	Yes	Yes	Codec interrupt(8 KHz)
TMRINT	0x000A	6th	Yes	Yes	Timer interrupt

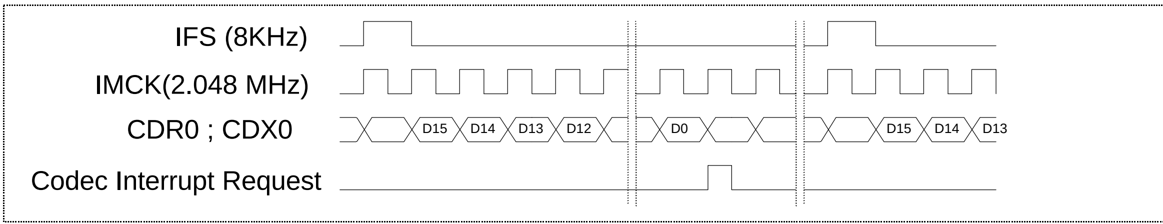
- **Interrupt Mask :** Each bit in I/O mapped register 4 (**IMR**) enables or disables the servicing of an individual interrupt. Global interrupt mask bit **INTM** equals “1” will mask all interrupt requests except reset and non-maskable interrupt request. **INTM** bit is set or reset by **DINT** or **EINT** instruction.
- NMI\ and INT1\ are edge triggered interrupt which request DSP during high to low transition.
- **Interruptible :** State that **INTM** or individual interrupt mask bits are in reset state (“0”) and no higher priority interrupt being serviced or exist in pending status.
- Program flow within repeat loop such like LUP, TBR ,MPA and SQRA instructions , and at time during DRAM data movement are all not interruptible.
- When a maskable interrupt request occurs , if DSP is in interruptible state, this request is granted by DSP and following service routine will be executed, otherwise this request will be hold in pending status bit until the DSP enters interruptible state again

- When DSP jumps into interrupt subroutine , **INTM** bit is automatically set high (After push status register onto stack) to prevent from nesting interrupt occurs. Execute **EINT** will change this situation and then make nesting interrupt permissible.
- Software hold state will be terminated and return to normal running if external interrupt or timer interrupt occurs and is granted by DSP.
- Single step(I/O mapped 7) provides an "always exist " interrupt condition. DSP will be interrupted after every instruction cycle.(DSP must be in interruptible state)
- No register will be automatically saved in stack register except status register during interrupt service routine. Cares should be taken with the current values stored in X-register, product register and accumulator , backup them at first in interrupt routine if needed.

3.4 APPLICATION INTERFACE UNIT
3.4.1 CODEC INTERFACE

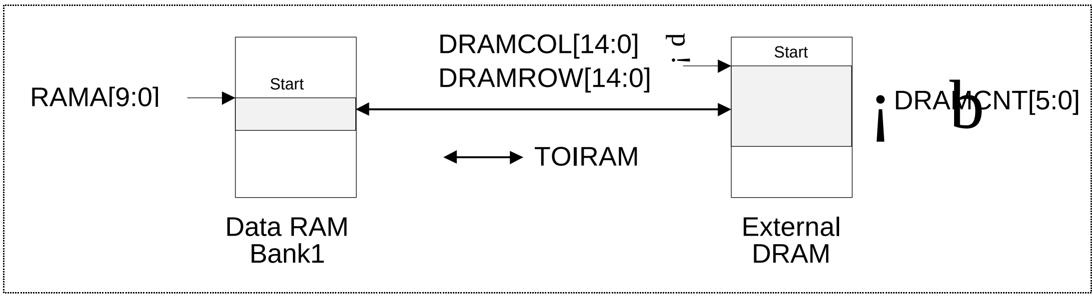


- IMCK is directly from PLL output. IFS equals to IMCK/256.



- After CFS positive pulse DSP begins to exchange data with external codec through CDR0/CDR1 and CDX0/CDX1. DSP transmits data at CMCK rising edge and receives data at CMCK falling edge.
- First data received will be put into the MSB of codec receive registers(I/O mapped 16,17).
- First data transmitted will come from the MSB of codec transmit registers(I/O mapped 16,17).
- After LSB (16th) data transmitted or received, DSP will generate an internal codec interrupt request.

3.4.2 DRAM INTERFACE



- DRAM controller support data movement between DSP RAM bank1 and external DRAM
- Support **FAST-PAGE** and **EDO-PAGE** mode DRAMs
- Data movement starts from non-zero value written to DRAMCNT[5:0] (I/O mapped 9)
- DSP will be hold during this data movement
- RAMA[9:0] (I/O mapped 9) specifies the starting address where data movement begin
- DRAMCOL[14:0](I/O mapped 10) and DARMROW[14:0](I/O mapped 11) specify the column and row part of DRAM starting address where this data movement begin
- TOIRAM(I/O mapped 11) defines the direction of data movement

  0 : DSP ¡ ÐDRAM 1: DSP ¡ ôDRAM
- DRAMSIZE[1:0](I/O mapped 8) define the configuration of DRAM data width :

  0 : x1 1: x4 2: x8 3: x16
- DRAMWAIT[2:0](I/O mapped 8) are the wait state number during DRAM data access

  Find the larger one of DRAMWAIT[2:0] below
  TRAC < 73 ns + (12.2 ns x DRAMWAIT[2:]) or
  TCAC < 11 ns + (12.2 ns x DRAMWAIT[2:0]
- Refresh mode : CAS before RAS refresh

Refresh cycle time : every 15.258 us (64 KHz)

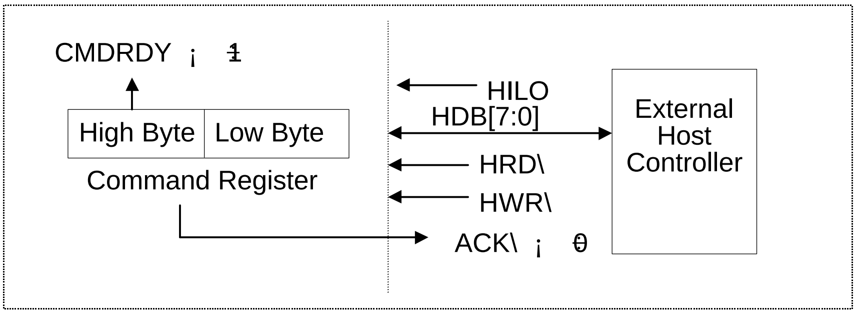
3.4.3 I/O FUNCTION

- **IN/OUT** instructions transfer data between internal data RAM and I/O mapped registers
- Up to 9 input port pins , 24 output port pins and 8 programmable bi-directional I/O pins can be used in general I/O function
- **HOSTM** is software bits in **CTLR**(I/O mapped 7).**DFS** is software bits in **EXCTLR**.
- IPT[3:0] built with internal pull high register(R ~ = 150 K ohm)
- XF\ can be used as general output pin which can be set or reset directly by RXF and SXF instruction

Application	HOSTM	DFS	Input Ports	Output ports	Bidirection I/O
External Host/DRAM	0	0	IPT[7:0]	OPT[15:0]	None
External Host/FLASH	0	1	IPT[8:0]	OPT[21:19;15:0]	None
No external Host/DRAM	1	0	IPT[7:0]	OPT[18:0]	BIO[7:0]
No external Host/FLASH	1	1	IPT[8:0]	OPT[21:0]	BIO[7:0]

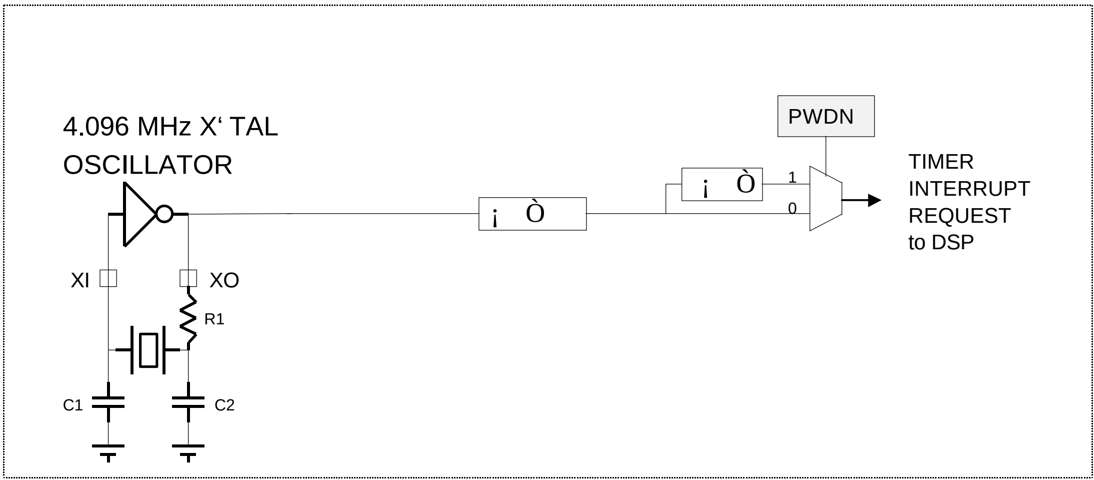
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG0	OPT[15:0]															
REG1(W)	OPT[23:19]															
REG1(R)							IPT[8:0]									
REG2	BIO[15:8]							BIO[7:0]								
REG7		OPT[18:16]														

3.4.4 HOST INTERFACE



- HOSTM (I/O mapped 7) define the HOST mode and multiplex some DSP I/O pins
  HOSTM : 0 : External host controller
  1 : No external host controller
- External host can read or write byte-wide command from or to this COMMAND REGISTER through HDB[7:0] and HILO select pins. HILO pin=1 will select upper byte of this register.
- When external host writes command to high byte of this register , **CMDRDY** bit in **CTLR** will be set till this register being read by DSP.
- When DSP writes command to this register , ACK\ pin will go low till high byte of this register being read by external host.

3.4.5 TIMER



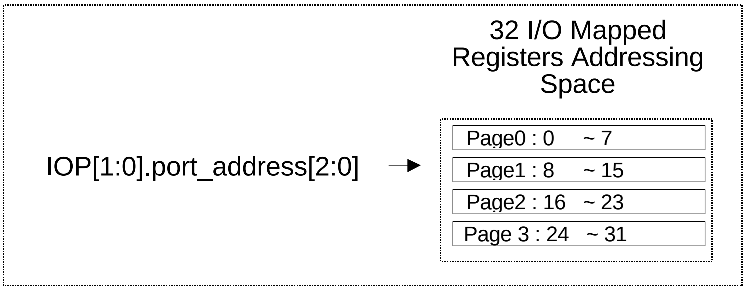
PWDN	Timer Interrupt Period
0	1 mili second
1	1/32 second

- Timer accuracy is determined by crystal' s character ,R1,C1,C2 and stray capacitance on PCB.

4.1 I/O Mapped Registers Summary

Register Name	Bit Width	I/O Address	Related Instructions	Descriptions
OPTR	16	0 (R/W)	IN/OUT	Output ports register
EXTOPTR	5	1 (R/W)	IN/OUT	Extended output ports register
IPTR	11	1 (R)	IN	Input ports register
BIOR/CMDR	16	2 (R/W)	IN/OUT	Bi-directional I/O ports / HOST command register
SHFCR	4	3 (R/W)	IN/OUT/SFL/SFR/SFRS	Shift count register
IMR	4	4 (R/W)	IN/OUT	Interrupt mask register
CDCMR	2	5 (R/W)	IN/OUT	Codec command register
CTLR	15	7 (R/W)	IN/OUT	Control register
WSTR	11	8 (R/W)	IN/OUT	Memory wait state and DRAM configuration register
DRAMACR	16	9 (R/W)	IN/OUT	DRAM access control register
DRAMCOLR	15	10 (R/W)	IN/OUT	DRAM column address register
DRAMROWR	16	11 (R/W)	IN/OUT	DRAM row address register
RCR	7	12 (R)	IN	Repeat counter
MODR	7	13(R)	IN/MOD/MODK	Modulo register for modulo addressing
XR	16	14 (R)	IN	X register (one of source registers to 16x16 multiplier)
SPR	5	15 (R)	IN/PSH/PSHH/PSHL POP/POPH/POPL	Stack pointer register
CDRR0	16	16 (R)	IN	Codec0 receive register
CDXR0	16	16 (W)	OUT	Codec0 transmit register
CDRR1	16	17 (R)	IN	Codec1 receive register
CDXR1	16	17 (W)	IOUT	Codec1 transmit register
PRODLR	15	18 (W)	OUT	Lower word of product register
PRODHR	16	19 (W)	OUT	Upper word of product register
TESTR	4	20 (W)	OUT	Testing register for internal use
PLLMR	5	21(R/W)	IN/OUT	PLL multiplication register
EXTCTLR	4	24(R/W)	IN/OUT	Extended Control register

Notes: (R) : This register is read only
(W) : This register is write only
(R/W) : This register can be read or write



PAGED I/O MAPPED REGISTER ADDRESSING

- Address of I/O mapped registers are composed of 2 bit I/O page pointer which are stored in status register and 3 bit within page port_address.
- **LIP** or **LIPK** instruction can be used to modify I/O page pointer, **SIP** and **SSS** instructions can be used to save I/O page pointer in data memory.
- 3 bit port_address are directly specified in part of instruction .

4.2 Non I/O Mapped Registers Summary

Register Name	Bit Width	I/O Address	Related Instructions	Descriptions
ACCH	16		SAH/ADH/SBH/POPH PSHH/AND/OR/XOR ABS/LAC ...	High word of accumulator
ACCL	16		SAL/ADL/SBL/POPL PSHL ...	Low word of accumulator
ACC	32		SBL/ADL/SFL/SFR NOM/ Multiply ...	32 bits accumulator
PC	16		CALL/CALA/TRAP/BS BZ/BACC/RET/RETI Reset/Interrupt	Program counter. Acts as program memory pointer
SSR	16		SSS/BS/BZ INTM : EINT/DINT TB : BIT OVM : ROVM/SOVM ARP : MAR IOP : LIP/SIP DP : LDP/SDP	Status register
AR0 ~ AR7	16x8		LAR/SAR/MAR	Auxiliary registers. Used as data memory pointer in register-indirect mode addressing

4.3 I/O Mapped Registers Description

4.3.1 OPTR (I/O mapped 0 : R/W) : Output Ports Register

Bit	Field	Default	Description
15 ~ 0	OPT[15:0]	0	Output ports register. Content of this register will be reflected to corresponding output pins.

4.3.2 EXTOPTR (I/O mapped 1 : W) : Extended Output Ports Register

Bit	14~ 11
Field	OPT[22:19]

Bit	Field	Default	Description
14 ~ 11	OPT[22:19]	0	Output ports register. Content of this register will be reflected to corresponding output pins.

4.3.3 IPTR (I/O mapped 1 : R) : Input Ports Register

Bit	10	9	8	7~0
Field	ACK\ / XF\	EROM	IPT8	IPT[7:0]

Bit	Field	Default	Description
10	ACK\ / XF\	1	Host acknowledge / External flag. Status bit, mapped from pin number 14.
9	EROM	X	Status bit, mapped from pin number 97
8	IPT8	X	Input port, mapped from pin number 93 when **DFS** bit in **EXTCTLR** equals one
7 ~ 0	IPT[7:0]	X	Input ports, mapped from 15 ~ 22

4.3.4 CMDR / BIOR (I/O mapped 2 : R/W) : Command or Bi-directional I/O ports Register

Bit	15 ~ 0	Option
Field	CMD[15:0]	If **HOSTM** bit in **CTLR** =0
Field	BIO[15:0]	If **HOSTM** bit in **CTLR** =1

4.3.4 CMDR / BIOR (I/O mapped 2 : R/W) : Command or Bi-directional I/O ports Register (Continued)

Bit	Field	Default	Description
15 ~ 0	CMD[15:0]	0	Parallel host command register. External Host can read or write byte-wide command from or to this **CMD** register through **HDB[7:0]** and **Hilo** select pins. **Hilo** pin =1 will select upper byte of this CMD register. Related Flag : When external host writes command to high byte of this register , **CMDRDY** bit in **CTLR** will be set till this register being read by DSP. When DSP writes command to this register , **ACK** pin will go low till high byte of this register being read by external host.
15 ~ 0	BIO[15:0]	0	BIO[7:0] are programmable bi-directional I/O ports. Ports direction of BIO[7:0] are programmed by BIO[15:8] bits respectively. BIO15 : 0 → BIO7 Input port 1 → BIO7 output port ....

4.3.5 SHFCR (I/O mapped 3 : R/W) : Shift Count Register

Bit	Field	Default	Description
4 ~ 0	SHFC[4:0]	0	Shift Count of barrel shifter . If the value of operand specified in SFL/SFR/SFRS usage equal 0 , left shift or right shift count of barrel shifter will be decided by this register. In normalize operation by "**NOM" ,** left shift counts are also stored in this register, but with one more extra bit for 31-bit shifting. In this case, the 5-bit SHFC[4:0] can be read by "IN" instruction for later operation. But for "OUT" instruction, only SHFC[3:0] can be written, SHFC[4] is always forced to 0 for backward compatible issue.

4.3.6 IMR (I/O mapped 4 : R/W) : Interrupt Mask Register

Bit	3	2	1	0
Field	SSM	TMRM	CODECM	INT1M

Bit	Field	Default	Description
			Interrupt mask bit will disable or enable individual interrupt 1 : disable 0 : enable
3	SSM	1	Single step interrupt mask bit
2	TMRM	1	Timer interrupt mask bit
1	CODECM	1	Codec interrupt mask bit
0	INT1M	1	External interrupt 1 mask bit

4.3.7 CDCMR (I/O mapped 5 : R or R/W) : Codec command register

Bit	9	8	2	1	0
Field	CDREADYX	ISDATAR	ICPDX	ISDENX	ISDATAW

Bit	Field	Default	R/W	Description
9	CDREADYX		R	If CDREADYX = 0, CODEC is ready
8	ISDATAR		R	DSP read register from CODEC
2	ICPDX	0	R/W	Set CODEC powerdown
1	ISDENX	1	R/W	DSP read/write register enable
0	ISDATAW	0	R/W	DSP write register to CODEC

4.3.8 CTLR (I/O mapped 7 : R/W) : Control Register

Bit	14 ~ 12	11	10	9	8
Field	OPT[18:16]	PWDN	SWHOLD		

Bit	7	6	5	4	3	2	1	0
Field		CMDRDY	PWDNS	SS		SNSEL	HOSTM	

Bit	Field	Default	Description
14 ~ 12	OPT[18:16]	0	Output ports to pin when **HOSTM** in **CTLR** = 1 . Share pin location with HILO , HRD\ and HWR\ .
11	PWDN	0	Power down mode enable. When power down mode being enabled by setting this bit DSP will switch running clock source to low x' tal, and turn off high X' tal or FLL to save power. When DSP is waken up by clearing this bit, DSP will stay in slow speed running for 62.5 ms till PLL output stabilize or high X' tal startup and stabilize , then switch back to high speed running. 1 = Power down.
10	SWHOLD	0	Software hold enable. When this bit being set , DSP will stop program execution, but high X' tal and PLL will not be turn off. Timer clock are still active during this mode. Software hold does not save more power like power down mode does, but responds faster for DSP resuming normal running from this mode when **SHOLD** bit is cleared by interrupt request. (Individual interrupt mask bit should be enabled first.) 1 = Hold.

4.3.8 CTLR (I/O mapped 7 : R/W) : Control Register (Continued)

6	CMDRDY	0	Host command ready flag. This bit will be set if external host write command to high byte of **CMDR** , and will be cleared when DSP reads **CMDR**.
5	PWDNS	0	Power down status bit. This bit will be set if **PWDN** bit being set. But will be cleared late by 62.5 ms after **PWDN** being cleared. This bit indicates what kind of speed DSP running with currently.
4	SS	0	Single step interrupt enable. When this bit being set , DSP will enter single step interrupt vector 0x0004 at end of each instruction.
2	SNSEL	0	Sign extended mode select in ADL/ADLL SBL/SBLL instructions. 0 : Fill "0" in upper word of accumulator. 1 : Sign extended in upper word of accumulator.
1	HOSTM	0	Host mode select : 0 : External host controller. 1 : No external host controller. This bit also acts as pins multiplex select. 0 : HDB[7:0] HILO HRD\ HWR\ ACK\ (External host) 1 : BIO[7:0] OPT18 OPT17 OPT16 XF\ (Internal host)

4.3.9 WSTR (I/O mapped 8 : R/W) : Memory Wait State Number and DRAM Configuration Register

Bit	10 ~ 9	8 ~ 6	5 ~ 3	2 ~ 0
Field	DRAMSIZE[1:0]	DATAWAIT[2:0]	DRAMWAIT[2:0]	PROGWAIT[2:0]

Bit	Field	Default	Description
10 ~ 9	DRAMSIZE[1:0]	1	DRAM configuration select. 0 : x1 1: x4 2: x8 3: x16
8 ~ 6	DATAWAIT[2:0]	7	External data memory wait state number. TAA or TCS < 26.5ns + (31 ns x DATAWAIT[2:0])
5 ~ 3	DRAMWAIT[2:0]	7	DRAM wait state number. Find the larger one below : TRAC < 73 ns + (15.5 ns x DRAMWAIT[2:0]) or TCAC < 11 ns + (15.5 ns x DRAMWAIT[2:0])
2 ~ 0	PROGWAIT[2:0]	7	External program memory wait state number. TAA or TCS < 26.5ns + (31 ns x PROGWAIT[2:0])
			All calculation is based on the assumption that DSP is running with 40.48 MHz Clock. If **FAST** bit in **EXTCTLR** equals 0 , all wait state numbers should be increased by one to meet the timing requirement stated above .

4.3.10 DRAMACR (I/O mapped 9 : R/W) : DRAM Access Control Register

Bit	15 ~ 10	9 ~ 0
Field	DRAMCNT[5:0]	RAMA[9:0]

Bit	Field	Default	Description
15 ~ 10	DRAMCNT[5:0]	0	Write a non zero value to this register will start data movement between internal data RAM and external DRAM .At this moment , DSP will hold operation till this data movement complete and these bits (DRAMCNT[5:0]) will be clear . DRAMCNT[5:0] indicate how many DRAM address location will involved in this movement.
9 ~ 0	RAMA[9:0]	0	RAM bank 1 OFFSET address. This address points to starting location where data movement begin. Data in extended RAM bank1(0x0800 ~ 0x09FF) can' t be moved.

4.3.11 DRAMCOLR (I/O mapped 10 : R/W) : DRAM Column Address Register

Bit	Field	Default	Description
14 ~ 0	DRAMCOL[14:0]	0	Column part of DRAM starting address where data movement begin

4.3.12 DRAMROWR (I/O mapped 11 : R/W) : DRAM Row Address Register

Bit	15	14 ~ 0
Field	TOIRAM	DRAMROW[14:0]

Bit	Field	Default	Description
15	TOIRAM	0	Data movement direction. 0 : Internal RAM of DSP → External DRAM 1 : External DRAM → Internal RAM of DSP
14 ~ 0	DRAMROW[14:0]	0	Row part of DRAM starting address where data movement begin

4.3.13 RCR (I/O mapped 12 : R) : Repeat Counter Register

Bit	Field	Default	Description
9 ~ 0	RC[9:0]	0	Instruction execution repeat counter. Affected instructions: TBR MPA SQRA and instructions within program loop . Read only register, but can be written by RPT and RPTK instructions. Real repeat number is RC[9:0]+1 .

4.3.14 MODR (I/O mapped 13 : R) : Modulo Register

Bit	Field	Default	Description
9 ~ 0	MOD[9:0]	0	Modulo Register. Non zero value of this register will enable modulo arithmetic in register-indirect addressing mode. A circular buffer , whose length is MOD[9:0]+1, will be formed. This circular buffer starts on K-word boundaries , where K is the smallest power of two that is equal to or greater than the size of the circular buffer. In register-indirect addressing mode operation, whenever the current auxiliary register points to the boundary of this circular buffer , it will be wrapped to the other side of the boundary for next address.

4.3.15 SPR (I/O mapped 15 : R) : Stack Register Pointer

Bit	Field	Default	Description
4 ~ 0	SP[4:0]	0	Stack register pointer. This pointer always points to the location within stack register where next data will be put. No pointer overflow or underflow protection built in, when such cases occur, the pointer will be wrapped to the other side of the stack.

4.3.16 CDRR0 (I/O mapped 16 : R) : First Codec Receive Register

Bit	Field	Default	Description
15 ~ 0	CDR0[15:0]	Undefined	After Codec frame sync. goes high, DSP begins to receive data from external Codec when Codec master clock goes low. The first data received will be put into the MSB of this register. When DSP has received sixteen bits data, DSP will stop receiving operation and trigger internal Codec interrupt.

4.3.17 CDXR0 (I/O mapped 16 : W) : First Codec Transmit Register

Bit	Field	Default	Description
15 ~ 0	CDX0[15:0]	Undefined	After Codec frame sync. goes high, DSP begins to transmit data to external Codec when Codec master clock goes high. The first data transmitted will come from the MSB of this register. When DSP has transmitted sixteen bits data, DSP will stop transmitting operation and trigger internal Codec interrupt.

4.3.18 CDRR1 (I/O mapped 17 : R) : Second Codec Receive Register

Bit	Field	Default	Description
15 ~ 0	CDR1[15:0]	Undefined	After Codec frame sync. Goes high, DSP begins to receive data from external Codec when Codec master clock goes low. The first data received will be put into the MSB of this register. When DSP has received sixteen bits data , DSP will stop receiving operation and trigger internal Codec interrupt.

4.3.19 CDXR1 (I/O mapped 17 : W) : Second Codec Transmit Register

Bit	Field	Default	Description
15 ~ 0	CDX1[15:0]	Undefined	After Codec frame sync. goes high, DSP begins to transmit data to external Codec when Codec master clock goes high. The first data transmitted will come from the MSB of this register. When DSP has transmitted sixteen bits data, DSP will stop transmitting operation and trigger internal Codec interrupt.

4.3.20 TESTR (I/O mapped 20 : W) : Test Register

Bit	Field	Default	Description
5 ~ 2	TEST[5:2]	0	Test bits used in testing.

4.3.21 PLLMR (I/O mapped 21 : W) : PLL Multiplication Factor Register

Bit	Field	Default	Description
12 ~ 0	FLLM[4:0]	0x14	PLL multiplication factor register. F_DSP = 4.096MHz / 2 * PLLM[4:0] Default: F_DSP = 4.096MHz / 2 * 20 = 40.96 MHz Range : 24.5 MHz < F_DSP < 49 MHz Lock in time ~= 10 ms Jitters : meet the requirement for digital answering machine application. For other applications ,care need to be taken.

4.3.22 EXTCTLR (I/O mapped 24 : R/W) : Extended Control Register

Bit	15 ~ 2	1	0
Field	Reserved	DFS	FAST

Bit	Field	Default	Description
15 ~ 2	Reserved	0	Output ports register. Content of this register will be reflected to corresponding output pins.
1	DFS	0	**DRAM** or **FLASH** interface pins select. This bit is used to multiplex pins between **DRAM** and **FLASH** interface. 0 : CAS\ DRD\ DWR\ RAS\ for DRAM interface 1: OPT21 OPT20 OPT19 IPT8 for FLASH interface
0	FAST	1	Output pad slew rate control bit. Used to reduced EMI. 0: Slow slew rate 1: Fast slew rate Affected Pins : ED[15:];EAD[15:0];EPCE\;EDCE\;EWR\;ERD\; RAS\;CAS\;DRD\;DWR\; Notes: Reset / set this bit affect wait state number requirement for external memory access.

4.4 NON I/O mapped registers Description

4.4.1 ACCH : Upper Word of Accumulator

Bit	Field	Default	Description
31 ~ 16	ACC[31:16]	Undefined	Upper word of accumulator.

4.4.2 ACCL : Lower Word of Accumulator

Bit	Field	Default	Description
15 ~ 0	ACC[15:0]	Undefined	Lower word of accumulator.

4.4.3 ACC : Accumulator

Bit	Field	Default	Description
31 ~ 0	ACC[31:0]	Undefined	Accumulator.

4.4.4 PC : Program Counter

Bit	Field	Default	Description
15 ~ 0	PC[15:0]	0x0000	Program counter. This counter is used as program memory pointer to control the DSP program flow. In **MPA** instruction , this counter is used as one of data memory pointer.

4.4.5 SSR : Status Register

Bit	15	14	13	12	11	10	9	8 ~ 6	5 ~ 4	3 ~ 0
Field	INTM	ARZ	SGN	OV	ACZ	TB	OVM	ARP[2:0]	IOP[1:0]	DP[3:0]

Bit	Field	Default	Description
			This register will be saved automatically in stack register when interrupt service begins and will be restored back when interrupt service has completed. **SSS** instruction can store this register to data memory. Some other instructions can modify or store part of this register.
15	INTM	1	Global interrupt mask bit. This bit can be set by **DINT** or reset by **EINT** instruction. Every time when DSP runs into interrupt service routine , this global mask bit will be set to disable any other interrupt. Clear this bit or execute EINT instruction can enable interrupt again and make nesting interrupt possible.
14	ARZ	1	This bit registers the last operated auxiliary register ' s value equal zero.
13	SGN	Undefined	MSB of high accumulator.
12	OV	0	Overflow flag for last ACCH operation. This flag will be cleared by any instructions which will generate result in accumulator.
11	ACZ	1	Accumulator zero flag. This bit reflects current accumulator status.

4.4.5 SSR : Status Register (Continued)

10	TB	0	Tested bit. This bit is used to stored one bit from data memory by **BIT** instruction, and will be tested by following **BZ** or **BS** instruction .
9	OVM	0	Overflow mode select. 0 : Disable overflow mode 1 : Enable overflow protection during arithmetic and shift left operation. This bit can be reset/set by **ROVM** / **SOVM** instruction.
8 ~ 6	ARP[2:0]	0	Auxiliary register pointer. This pointer points to one of eight auxiliary registers as current ar in register-indirect addressing mode.
5 ~ 4	IOP[1:0]	0	I/O mapped register Page pointer. This DSP can access total 32 internal I/O ports address formed by 4 pages , each page contains eight ports address . Port address is specified as immediate operand in IN / OUT instruction. These bits can be modified by **LIP/LIPK** or saved by **SIP** instructions.
3 ~ 0	DP[3:0]	0	Internal data memory page pointer used in direct memory addressing mode. The DSP contains total 16 pages of internal RAM whose range is from 0x0000 to 0x07FF.Each page contain 128 words, the words address within page can be specified as immediate operand in related instructions. These bits can be modified by **LDP/LDPK** or saved by **SDP** instructions. Internal RAM located within (0x0800 ~ 0x09FF) can be accessed only by register-indirect mode.

4.4.6 STR : Stack Register

Bit	15 ~ 0
Field	ST[31:0][15:0]

Bit	Field	Default	Description
32x16	STACK REGISTER	Undefined	This register is used to stored return address from program counter in the case of interrupt service begin or CALL/CALA instruction execution.Or acts as data buffer for use in data exchange with **ACCH, ACCL** , **SSR** and data from memory

4.4.7 ARS : Auxiliary Registers

Bit	15 ~ 0	15 ~ 0	15 ~ 0	15 ~ 0	15 ~ 0	15 ~ 0	15 ~ 0	15 ~ 0
Field	AR7[15:0]	AR6[15:0]	AR5[15:0]	AR4[15:0]	AR3[15:0]	AR2[15:0]	AR1[15:0]	AR0[15:0]

These Auxiliary registers are used as data memory pointer in register-indirect mode addressing .
ARP[2:0] in status register will choose one of them as **current AR** , and the current AR acts as data memory pointer in related instruction operation.
Content of current ar can be modify as follow :

ar + 0 → ar or
ar + 1 → ar or ar - 1 → ar
ar + 2 → ar or ar - 2 → ar
ar + ar0 → ar or ar - ar0 → ar

ARP[2:0] can also be updated with new auxiliary register pointer : narp → arp
All operations stated above work in parallel with instruction execution.

5.1 INSTRUCTION SET SUMMARY :

DATA UNIT INSTRUCTIONS:1.Arithmetic 2.Logic/Shift 3.Data movement 4.Mode setting	
Mnemonic	**Description**
ADH/ADHK/ADHL	Add data (from memory) or constant to high accumulator
ADL/ADLK/ADLL	Add data (from memory) or constant to low accumulator
SBH/SBHK/SBHL	Subtract data (from memory) or constant from high accumulator
SBL/SBLK/SBLL	Subtract data (from memory) or constant from low accumulator
ABS	Absolute value of high accumulator
OR/ORK/ORL	OR data (from memory) or constant with high accumulator
AND/ANDK/ANDL	AND data (from memory) or constant with high accumulator
XOR/XORK/XORL	Exclusive-OR data (from memory) or constant with high accumulator
SFL/SFR/SFRS	Shift contents of accumulator left/right/right with sign extended
NOM	Normalize contents of accumulator
LAC/LACK/LACL	Load data (from memory) or constant to high accumulator
SAH/SAL	Store contents of high or low accumulator to data memory
POPH/POPL	Pop top of stack to high/low accumulator
PSHH/PSHL	Push high/low accumulator onto stack
SOVM/ROVM	Set/Reset overflow mode

AUXILIARY REGISTERS AND DATA/IO PAGE POINTER INSTRUCTIONS	
Mnemonic	**Description**
LAR/LARK/LARL	Load data (from memory) or constant to auxiliary register
SAR	Store auxiliary register to data memory
MAR	Modify auxiliary register pointer and update auxiliary register pointer
MOD/MODK	Load data (from memory) or constant to modulo register
LIP/LIPK	Load data (from memory) or constant to modify I/O page pointer in status register
SIP	Store I/O page pointer in status register to data memory
LDP/LDPK	Load data (from memory) or constant to modify data page pointer in status register
SDP	Store data page pointer in status register to data memory

5.1 INSTRUCTION SET SUMMARY : (Continued)

PROGRAM FLOW CONTROL INSTRUCTIONS	
Mnemonic	**Description**
RPT/RPTK	Load data (from memory) or constant to repeat counter
LUP/LUPK	Enable loop operation /Enable loop operation with short constant
BS/BZ	Branch immediate if bit being tested set or reset
CALA	Call subroutine indirectly specified by contents of high accumulator
CALL	Call subroutine
BACC	Branch to address specified by contents of high accumulator
DINT/EINT	Disable / Enable interrupt
RET/RETI	Return from subroutine / interrupt
NOP	No operation

DATA MOVEMENT AND MISCELLANEOUS INSTRUCTIONS	
Mnemonic	**Description**
OUT/OUTK/OUTL	Load data (from internal data memory) or constant to I/O mapped register
IN	Move data from I/O mapped register to internal data memory
BIT	Move data bit from memory to TB in status register
SSS	Store status register to data memory
POP/PSH	Pop top of stack to data memory/push data memory value onto stack
SXF/RXF	Set/reset external flag

5.2 ACRONYMS AND NOTATIONS :

→	Data transfer
pc	Program counter
pma16	16 bit program memory address
(arp)	auxiliary register pointed by arp. also called current auxiliary register
ar	current Auxiliary register
ar0	the first Auxiliary register
(arps)	Auxiliary register pointed by arps
dma7	7 bit direct memory address within data page
dma	16 bit whole data memory address formed by 0.dp(3:0).dma7 or by 16 bits **current ar**
(dma)	data memory pointed by dma
(dma)(3:0)	lowest nibble of (dma)
#	bit location indicator
(dma)(#cnst4)	bit data of (dma) which is pointed by cnst4
*	register indirect mode addressing operator , can be one of : +0 , + ,++ , - , --, +ar0 , -ar0
[,narp]	Next AR Pointer, narp is a 3 bit constant. "[" "]" means option , not real expression.
cnst2,cnst3,cnst4, cnst7,cnst16	2 bit , 3 bit , 4 bit , 7 bit , 16 bit constant
1(DI),2(DE)	one cycle for data memory internal , two cycles for data memory external
acc	accumulator
p	product register
sp	stack register pointer
(sp)	stack register pointed by **sp**
ss	status register
CTLR	control register
lc(2:0)	loop counter
mr(6:0)	modulo register
rc	repeat counter
norm(4:0)	normalize register
x	multiplication operator
dp(3:0)	data page pointer
iop(1:0)	I/O page pointer
port_address	3 bit address within I/O page
io_address	5 bit I/O address formed by iop(1:0) and port_address
(io_address)	I/O mapped register pointed by io_address

NOTE 1 : Instruction Encoding For Register Indirect Addressing

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE								1	E6	E5	E4	E3	E2	E1	E0

Operand	Encoding			Operation
*	E6	E5	E4	
+0	0	0	0	No operation
-AR0	0	0	1	(arp) -ar0 → (arp)
+AR0	0	1	0	(arp)+ar0→ (arp)
+	1	0	0	(arp) + 1 → (arp)
-	1	0	1	(arp) - 1 → (arp)
++	1	1	0	(arp) + 2 → (arp)
--	1	1	1	(arp) - 2 → (arp)

Operand	Encoding	Encoding	Operation
[,narp]	E3	E2 E1 E0	
None	0		No operation
, narp	1	narp	narp → arp

NOTE II : Instruction Encoding For Register Indirect Addressing

For MB , MBA , MBS Instructions only

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE													E2	E1	E0

Operand	Encoding			Operation
*	E2	E1	E0	
+0	0	0	0	No operation
-AR0	0	0	1	(arp) -ar0 → (arp)
+AR0	0	1	0	(arp)+ar0→ (arp)
+	1	0	0	(arp) + 1 → (arp)
-	1	0	1	(arp) - 1 → (arp)
++	1	1	0	(arp) + 2 → (arp)
--	1	1	1	(arp) - 2 → (arp)

5.3 INSTRUCTION SET DESCRIPTION

ABS	**Absolute value of high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ABS
Operation:	pc + 1 → pc
	|acc(31:16)| → acc(31:16)
Words:	1
Cycles:	1
Note:	|0x8000| will exceed the maximum positive number which can be represented . and will cause incorrect result .

ADH	**Add data from memory to high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ADH dma7
	ADH * [,narp]
Operation:	pc + 1 → pc
	acc(31:16) + (dma) → acc(31:16)
Words:	1
Cycles:	1(DI) , 2(DE)

ADHK	**Add immediate 7-bit unsigned short constant to high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ADHK cnst7
Operation:	pc + 1 → pc
	acc(31:16) + cnst7 → acc(31:16)
Words:	1
Cycles:	1

ADHL	**Add immediate 16-bit long constant to high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ADHL cnst16
Operation:	pc + 2 → pc
	acc(31:16) + cnst16 → acc(31:16)
Words:	2
Cycles:	2

ADL	**Add data from memory to low accumulator**	
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Syntax:	ADL dma7 ADL * [,narp]	
Operation:	pc + 1 → pc acc(31:0) + (dma) → acc(31:0)	
Words:	1	
Cycles:	1(DI) , 2(DE)	
Note:	Data operand is expanded into 32 bit long width with MSBs optionally sign extended or filled with "0" bits. This option is controlled by **SNSEL** bit in **CTLR**.	

ADLK	**Add immediate 7-bit unsigned short constant to low accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ADLK cnst7
Operation:	pc + 1 → pc acc(31:0) + cnst7 → acc(31:0)
Words:	1
Cycles:	1

ADLL	**Add immediate 16-bit long constant to low accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ADLL cnst16
Operation:	pc + 2 → pc acc(31:0) + cnst16 → acc(31:0)
Words:	2
Cycles:	2
Note:	Data operand is expanded into 32 bit long width with MSBs optionally sign extended or filled with "0" bits. This option is controlled by **SNSEL** bit in **CTLR**.

AND	**AND data from memory with high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	AND dma7
	AND * [,narp]
Operation:	pc + 1 → pc
	acc(31:16) **AND** (dma) → acc(31:16)
Words:	1
Cycles:	1(DI) , 2(DE)

ANDK	**AND immediate 7-bit short constant with high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ANDK cnst7
Operation:	pc + 1 → pc
	acc(22:16) **AND** cnst7 → acc(22:16)
	0 → acc(31:23)
Words:	1
Cycles:	1

ANDL	**AND immediate 16-bit long constant to high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	ADLL cnst16
Operation:	pc + 2 → pc
	acc(31:16) **AND** cnst16 → acc(31:16)
Words:	2
Cycles:	2

APAC	**Add product register to accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	APAC
Operation:	pc + 1 → pc
	acc + p → acc
Words:	1
Cycles:	1

BACC	**Branch to address specified by high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	BACC
Operation:	acc(31:16) → pc
Words:	1
Cycles:	2

BIT **Copy data bit from memory to TB bit in status register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: BIT dma7 , cnst4
BIT * , cnst4 [,narp]

Operation: pc + 1 → pc
(dma)(#cnst4) → **TB** bit in status register

Words: 1

Cycles: 1(DI) , 2(DE)

BS **Branch immediate if bit being tested equal one**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: BS cnst3 , pma16

Operation: If ss(#(cnst3+8)) = 1 then pma16 → pc
else pc+2 → pc

Words: 2

Cycles: 3

Note: BS instruction will test one bit of status register' s upper byte. Bit 15 is always one, and bit 8 is not defined in this test condition..

15	14	13	12	11	10	9	8
1	arz	sgn	ov	acz	tb	ovm	

: Part of upper byte of status register

BZ **Branch immediate if bit being tested equal zero**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: BS bbb , pma16

Operation: If ss(#(cnst3+8)) = 0 then pma16 → pc
else pc+2 → pc

Words: 2

Cycles: 3

Note: BS instruction will test one bit of status register' s upper byte. Bit 15 is always one, and bit 8 is not defined in this test condition..

15	14	13	12	11	10	9	8
1	arz	sgn	ov	acz	tb	ovm	

: Part of upper byte of status register

CALA **Call subroutine indirectly**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: CALA

Operation: pc + 1 → (sp)
sp + 1 → sp
acc(31:16) → pc

Words: 1

Cycles: 2

CALL **Call subroutine directly**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: CALL pma16

Operation: pc + 1 → (sp)
sp + 1 → sp
pma16 → pc

Words: 2

Cycles: 3

DINT **Disable interrupt**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: DINT

Operation: pc + 1 → pc
1 → **INTM** bit in status register

Words: 1

Cycles: 1

EINT **Enable interrupt**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: DINT

Operation: pc + 1 → pc
0 → **INTM** bit in status register

Words: 1

Cycles: 1

IN **Move data from I/O mapped register to internal data memory**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: IN dma7, port_address
IN * , port_address [,narp]

Operation: pc + 1 → pc
iop(1:0).port_address → io_address
(io_address) → (dma)

Words: 1

Cycles: 1

LAC **Load data from memory to high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LAC dma7
LAC * [,narp]

Operation: pc + 1 → pc
(dma) → acc(31:16)
0 → acc(15:0)

Words: 1

Cycles: 1(DI) , 2(DE)

LACK — **Load immediate 7-bit unsigned short constant to high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LACK cnst7

Operation: pc + 1 → pc
cnst7 → acc(22:16)
0 → acc(31:23)
0 → acc(15:0)

Words: 1

Cycles: 1

LACL — **Load immediate 16-bit long constant to high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LACL cnst16

Operation: pc + 2 → pc
cnst16 → acc(31:16)
0 → acc(15:0)

Words: 2

Cycles: 2

LAR — **Load data from memory to auxiliary register specified**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LAR dma7 , arps
LAR * , arps [,narp]

Operation: pc + 1 → pc
(dma) → (arps)

Words: 1

Cycles: 1(DI) , 2(DE)

Note: Data from memory(by direct mode or register indirect mode) will be loaded into auxiliary register specified in instruction encoding(arps : 3 bit constant). Post increment or decrement on current auxiliary register will not be performed during this instruction operation, but new arp can be changed.

LARK **Load immediate 7-bit short constant to auxiliary register specified**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LARK cnst7 , arps
Operation: pc + 1 → pc
cnst7 → (arps)(6:0) , 0 → (arps)(15:7)
Words: 1
Cycles: 1

LARL **Load immediate 16-bit long constant to auxiliary register specified**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LARL cnst16 ,arps
Operation: pc + 2 → pc
cnst16 → (arps)
Words: 2
Cycles: 2

LDP **Load data from memory to modify data page pointer in status register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LDP dma7
LDP * [,narp]
Operation: pc + 1 → pc
(dma)(3:0) → dp(3:0)
Words: 1
Cycles: 1(DI) , 2(DE)

LDPK **Load 4-bit short constant to modify data page pointer in status register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LDPK cnst4
Operation: pc + 1 → pc
cnst4 → dp(3:0)
Words: 1
Cycles: 1

LIP	**Load data from memory to modify I/O page pointer in status register**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	LIP dma7
	LIP * [,narp]
Operation:	pc + 1 → pc
	(dma)(5:4) → iop(1:0)
Words:	1
Cycles:	1(DI) , 2(DE)

LIPK	**Load 2-bit short constant to modify I/O page pointer in status register**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	LIPK cnst2
Operation:	pc + 1 → pc
	cnst2 → iop(1:0)
Words:	1
Cycles:	1

LUP	**Enable loop operation**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	LUP dma7, loop_number
	LUP * , loop_number [,narp]
Operation:	pc + 1 → pc
	(dma)(9:0) → rc(9:0)
	loop_number → lc(2:0)
Words:	1
Cycles:	1(DI) , 2(DE)
Note:	This instruction will enable hardware loop operation , and the following (loop_number+1) words instruction(program loop) will be executed repeatedly (rc +1) times.
	Branch and call instructions are not allowed within program loop.

LUPK **Load 7-bit short constant to repeat counter and enable loop operation**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: LUP cnst7 , loop_number

Operation: pc + 1 → pc
cnst7 → rc(6:0)
loop_number → lc(2:0)

Words: 1

Cycles: 1

Note: This instruction will enable hardware loop function , and the following (loop_number+1) words instruction (program loop)will be executed repeatedly (rc +1) times. No branch and call instructions are allowed within program loop.

MAR **Modify current auxiliary register and update auxiliary register pointer**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: MAR * [,narp]

Operation: pc + 1 → pc
Modify the content of current auxiliary register pointed by current auxiliary register pointer, and update this pointer with new pointer.

Words: 1

Cycles: 1

MOD **Load data from memory to modulo register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: MOD dma7
MOD * [,narp]

Operation: pc + 1 → pc
(dma)(9:0) → mr(9:0)

Words: 1

Cycles: 1(DI) , 2(DE)

MODK **Load immediate 7-bit short constant to modulo register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: MODK cnst7

Operation: pc + 1 → pc
cnst7 → mr(6:0)

Words: 1

Cycles: 1

NOP **No operation**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: NOP
Operation: pc + 1 → pc
Words: 1
Cycles: 1

NOM **Normalize the content of accumulator**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: NOM
Operation: pc + 1 → pc
Normalize(acc) → acc
Left Shift Count → **SHFC**
Words: 1
Cycles: 1
Note: This **NOM** instruction performs hardware normalization operation on signed two' s complement numbers stored in the accumulator. The left shifted counts during normalization are stored in shift count register(**SHFC**) .

OR **OR data from memory with high accumulator**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: OR dma7
OR * [,narp]
Operation: pc + 1 → pc
acc(31:16) **OR** (dma) → acc(31:16)
Words: 1
Cycles: 1(DI) , 2(DE)

ORK **OR immediate 7-bit short constant with high accumulator**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: ORK cnst7
Operation: pc + 1 → pc
acc(22:16) **OR** cnst7 → acc(22:16)
acc(31:23) → acc(31:23)
Words: 1
Cycles: 1

ORL **OR immediate 16-bit long constant with high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: ORL cnst16
Operation: pc + 2 → pc
acc(31:16) **OR** cnst16 → acc(31:16)
Words: 2
Cycles: 2

OUT **Load data from internal data memory to I/O mapped register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: OUT dma7, port_address
OUT * , port_address [,narp]
Operation: pc + 1 → pc
iop(1:0).port_address → io_address
(dma) → (io_address)
Words: 1
Cycles: 1

OUTK **Move 7-bit short constant to I/O mapped register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: OUTK cnst7, port_address
Operation: pc + 1 → pc
iop(1:0).port_address → io_address
cnst7 → (io_address)(6:0)
Words: 1
Cycles: 1

OUTL **Move 16-bit long constant to I/O mapped register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: OUTL cnst16, port_address
Operation: pc + 2 → pc
iop(1:0).port_address → io_address
cnst16 → (io_address)
Words: 2
Cycles: 2

POP **Pop top of stack to data memory**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: POP dma7
POP * [,narp]

Operation: pc + 1 → pc
sp - 1 → sp
(sp) → (dma)

Words: 1

Cycles: 1(DI) , 2(DE)

POPH **Pop top of stack to high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: POPH

Operation: pc + 1 → pc
sp - 1 → sp
(sp) → acc(31:16)

Words: 1

Cycles: 1

POPL **Pop top of stack to low accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: POPL

Operation: pc + 1 → pc
sp - 1 → sp
(sp) → acc(15:0)

Words: 1

Cycles: 1

PSH **Push data from memory onto stack**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: PSH dma7
PSH * [,narp]
Operation: pc + 1 → pc
(dma) → (sp)
sp + 1 → sp
Words: 1
Cycles: 1(DI) , 2(DE)

PSHH **Push high accumulator onto stack**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: PSHH
Operation: pc + 1 → pc
acc(31:16) → (sp)
sp + 1 → sp
Words: 1
Cycles: 1

PSHL **Push low accumulator onto stack**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: PSHL
Operation: pc + 1 → pc
acc(15:0) → (sp)
sp + 1 → sp
Words: 1
Cycles: 1

RET **Return from subroutine**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: RET
Operation: sp - 1 → sp
(sp) → pc
Words: 1
Cycles: 2

RETI **Return from interrupt**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: RETI
Operation: sp - 1 → sp
(sp) → pc
sp -1 → sp
(sp) → status register
Words: 1
Cycles: 2

ROVM **Reset overflow mode**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: ROVM
Operation: pc + 1 → pc
0 → **OVM** bit in status register
Words: 1
Cycles: 1

RPT **Load data from memory to repeat counter**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: RPT dma7
RPT * [,narp]
Operation: pc + 1 → pc
(dma)(9:0) → rc(9:0)
Words: 1
Cycles: 1(DI) , 2(DE)

RPTK **Load immediate 7-bit short constant to repeat counter**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: RPTK cnst7
Operation: pc + 1 → pc
cnst7 → rc
Words: 1
Cycles: 1

RXF	**Reset external flag**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	RXF
Operation:	pc + 1 → pc
	0 → **XF** but **1** → **XF** pin
Words:	1
Cycles:	1
Notes:	**XF\\ is an inverted output of XF .**

SAH	**Store content of high accumulator to data memory**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SAH dma7
	SAH * [,narp]
Operation:	pc + 1 → pc
	acc(31:16) → (dma)
Words:	1
Cycles:	1(DI) , 2(DE)

SAL	**Store content of low accumulator to data memory**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SAL dma7
	SAL * [,narp]
Operation:	pc + 1 → pc
	acc(15:0) → (dma)
Words:	1
Cycles:	1(DI) , 2(DE)

SAR	**Store content of auxiliary register specified to data memory**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SAR dma7 , arps
	SAR * , arps [,narp]
Operation:	pc + 1 → pc
	(arps) → (dma)
Words:	1
Cycles:	1(DI) , 2(DE)

SBH	**Subtract data (from memory) from high accumulator**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SBH dma7
	SBH * [,narp]
Operation:	pc + 1 → pc
	acc(31:16) - (dma) → acc(31:16)

Words: 1
Cycles: 1(DI) , 2(DE)

SBHK **Subtract immediate 7-bit unsigned short constant from high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SBHK cnst7
Operation: pc + 1 → pc
acc(31:16) - cnst7 → acc(31:16)
Words: 1
Cycles: 1

SBHL **Subtract immediate 16-bit long constant from high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SBHL cnst16
Operation: pc + 2 → pc
acc(31:16) - cnst16 → acc(31:16)
Words: 2
Cycles: 2

SBL **Subtract data (from memory) from low accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SBL dma7
SBL * [,narp]
Operation: pc + 1 → pc
acc(31:0) - (dma) → acc(31:0)
Words: 1
Cycles: 1(DI) , 2(DE)
Note: Data operand is expanded into 32 bit long width with MSBs optionally sign extended or filled with "0" bits. This option is controlled by **SNSEL** bit in **CTLR**.

SBLK **Subtract immediate 7-bit unsigned short constant from low accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SBLK cnst7
Operation: pc + 1 → pc
acc(31:0) - cnst7 → acc(31:0)
Words: 1
Cycles: 1

SBLL **Subtract immediate 16-bit long constant from low accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SBLL cnst16
Operation: pc + 2 → pc
acc(31:0) - cnst16 → acc(31:0)

Words:	2
Cycles:	2
Note:	Data operand is expanded into 32 bit long width with MSBs optionally sign extended or filled with “0” bits. This option is controlled by SNSEL bit in CTLR.

SDP **Store data page pointer in status register to data memory**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax:	SDP dma7
	SDP * [,narp]
Operation:	pc + 1 → pc
	dp(3:0) → (dma)(3:0)
Words:	1
Cycles:	1(DI) , 2(DE)

SFL	**Shift content of accumulator left (LSBs filled with zero)**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SFL cnst4

Operation:	pc + 1 → pc
	If (cnst4!= 0)
	then
	acc(31:0) will be shifted left by cnst4 bits
	else
	acc(31:0) will be shifted left by shfc[3:0] bits

Words:	1
Cycles:	1
Notes:	**shfc[3:0]** is the content stored in **SHFC** register (I/O mapped 3).

SFR	**Shift content of accumulator right (MSBs filled with zero)**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SFR cnst4

Operation:	pc + 1 → pc
	If (cnst4!= 0)
	then
	acc(31:0) will be shifted right by cnst4 bits
	else
	acc(31:0) will be shifted right by shfc[3:0] bits

Words:	1
Cycles:	1
Notes:	**shfc[3:0]** is the content stored in **SHFC** register (I/O mapped 3).

SFRS	**Shift content of accumulator right (MSBs filled with sign extended bit)**
	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Syntax:	SFRS cnst4

Operation:	pc + 1 → pc
	If (cnst4!= 0)
	then
	acc(31:0) will be shifted right by cnst4 bits
	else
	acc(31:0) will be shifted right by shfc[3:0] bits

Words:	1
Cycles:	1
Notes:	**shfc[3:0]** is the content stored in **SHFC** register (I/O mapped 3).

SIP **Store I/O page pointer in status register to data memory**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SIP dma7
SIP * [,narp]
Operation: pc + 1 → pc
iop(1:0) → (dma)(1:0)
Words: 1
Cycles: 1(DI) , 2(DE)

SOVM **Set overflow mode (Enable overflow mode protection)**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SOVM
Operation: pc + 1 → pc
1 → **OVM** bit in status register
Words: 1
Cycles: 1
Note: When **OVM** bit being set , overflow mode protection is enabled. IF result of data operation during add/subtract and shifting instructions execution exceed the maximum or minimum value that can be represented by the accumulator , data in accumulator will be saturated to the largest positive or negative number that can be represented.(0x7FFFF FFFF or 0x8000 0000)

SSS **Store content of status register to data memory**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SSS dma7
SSS * [,narp]
Operation: pc + 1 → pc
ss(15:0) → (dma)
Words: 1
Cycles: 1(DI) , 2(DE)

SXF **Set external flag**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: SXF
Operation: pc + 1 → pc
1 → XF but **0 → XF** pin
Words: 1
Cycles: 1
Notes: **XF\ is an inverted output of XF .**

XOR | **XOR data from memory with high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: XOR dma7
XOR * [,narp]

Operation: pc + 1 → pc
acc(31:16) **XOR** (dma) → acc(31:16)

Words: 1

Cycles: 1(DI) , 2(DE)

XORK | **XOR immediate 7-bit short constant with high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: XORK cnst7

Operation: pc + 1 → pc
acc(22:16) **XOR** cnst7 → acc(22:16)
acc(31:23) → acc(31:23)

Words: 1

Cycles: 1

XORL | **XOR immediate 16-bit long constant with high accumulator**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Syntax: XORL cnst16

Operation: pc + 2 → pc
acc(31:16) **XORL** cnst16 → acc(31:16)

Words: 2

Cycles: 2

6 PCM CODEC

6.1 CODEC OVERVIEW

The PCM CODEC integrates key functions of the analog-front-end of DAM (with Digital Speakerphone) related products into an integrated circuit. The PCM CODEC is especially powerful when applied to some DAM models which are intended to meet different countries' specifications in the same system hardware. User can achieve this goal by simply setting control firmware. This benefit will help DAM system makers to save developing time and R&D resources.

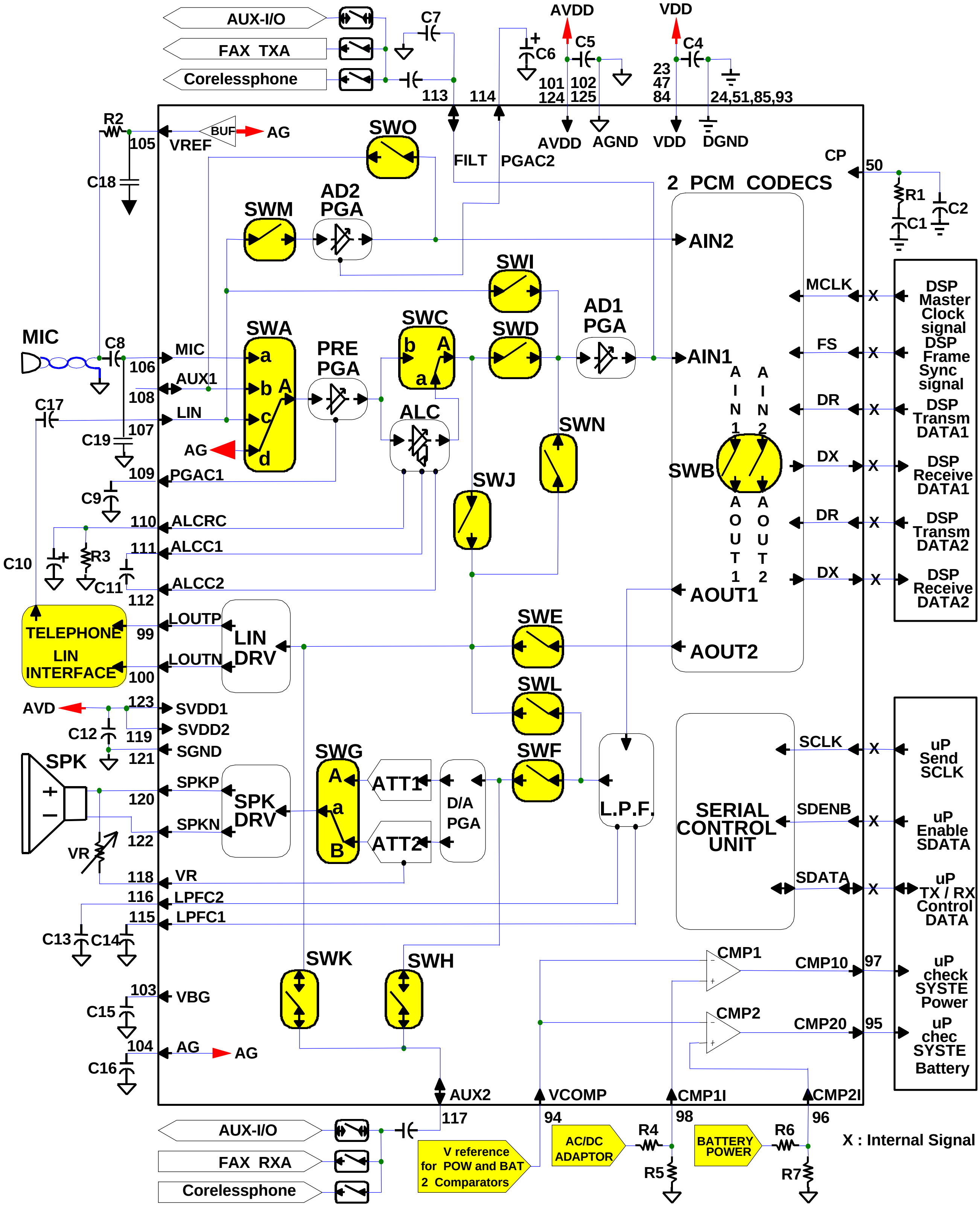
The built-in CODEC has two A/D, D/A converters so as to meet the requirement of the digital speakerphone application. The on-chip digital filters, which are carried out with 16-bit and 2's complement format, are used to get required frequency response of a PCM CODEC. PCM CODEC is 16-bit format with 14-bit resolution.

Before the A/D digitizing the voice-band analog signal into digital format, the analog signal can be processed by a built-in Automatic Level Control (***ALC***) and PRE-Programmable Gain Amplifier (***PRE-PGA***). The ***ALC*** circuit controls the signal level about 1.2Vpp and ***AD1-PGA*** can provide 0 ~ 18dB gain to get more larger signal. The ***PRE-PGA*** circuit is used to control the gain of different sources like **MIC**, **AUX1** or **LIN** input.

After the digital data is converted into analog signal by D/A converter, a fully differential line driver and speaker driver are supported to drive the telephone line and 8Ω speaker directly without needing any external amplifiers. Besides, the analog signal can be monitored by passing the on-chip volume control or external volume control.

The MX93132 supports many switches as well. User can program the control registers of the PCM CODEC to accomplish all specific operations of DAM (with digital speakerphone function) related products.

BLOCK DIAGRAM (PCM CODEC)



BASIC COMPONENTS REQUIRED

REFERANCE	PART	DESCRIPTION
***R1**	68KΩ	the resistor for internal PLL charge pump circuits
R2	2KΩ	current limit resistor; to limit MIC bias current, please follow MIC specification
R3	560KΩ	***ALC*** release time constant; see **FIG. 10**
R4, R5		to scale down DC power supply (**CMP1I**) for reference to **VCOMP** to check power low
R6, R7		to scale down battery power (**CMP2I**) for reference to **VCOMP** to check battery low
***C1**	100pF	the capacitor for internal PLL charge pump circuits
***C2**	6pF	the capacitor for internal PLL charge pump circuits
C8, C17	0.1uF	DC blocking capacitor (0.1~10uF)
C11	0.22uF	DC blocking capacitor (0.1~10uF); H.P.F. 3dB point : fc ¡ Ü1/2 £ k 4.4KΩ ∗ C6 (0.22uF) = 164Hz
C6	10uF	DC offset canceling compensative capacitor (4.7~10uF, the larger the better)
C9	0.1uF	DC offset canceling compensative capacitor (0.1~1uF, the larger the better)
C3, C4, C5, C12, C16	0.1uF	De-couple capacitor (0.1~10uF)
C15	0.1uF	De-couple capacitor (0.01~10uF); see **FUNCTIONAL DESCRIPTION**
C10	10uF	***ALC*** attack time constant; see **FIG. 9**
***C7**	5000pF	anti-aliasing capacitor
C13, C14		passive ***L.P.F.***; 3dB point : fc ¡ Ü1/2 £ k 3KΩ ∗ C13 (where C13 = C14)
***VR1**	10KΩ	to attenuate the input signal from ***SWH*** or ***SWF***, if use digital volume control, then do not need a resistor between **VR** and **SPKP**

@ where : " ***** " mark shows the requirement of the component can not be changed.

6.2 FUNCTIONAL DESCRIPTION

. PCM CODEC

. The block includes ***A/D*** & ***D/A*** converters and all digital filters;

1. ***A/D*** & ***D/A*** Converters

A/D Channel :

A. Input Range : 0 ~ 3Vpp (3Vpp as A/D 0dB full swing (0dBFS));

B. Digital Filters : For the purpose of out-of-band noise filtering, IIR digital filters are implemented on the same chip (>26dB / 60Hz; <1dB / 300Hz ~ 3.4KHz; >14dB / 3.6KHz ~ 4.6KHz; >32dB / 4.6KHz);

D/A Channel :

A. Output swing : 0 ~ 3Vpp (3Vpp as D/A 0dB full swing (0dBFS));

B. Digital Filters :

a. The digital input applied to D/A converter can not be a DC signal other than idle (bits all zero), as limit cycles in the embodiment method at a level of -70dBm will present at the analog output.

2. Data format: Linear format

@ Linear 16-bit format : 14-bit resolution with 2 LSB = 0

SIGN \ SCALE	MIN	MAX
POSITIVE	0000 0000 0000 0000	0111 1111 1111 1100
NEGATIVE	1111 1111 1111 1100	1000 0000 0000 0000

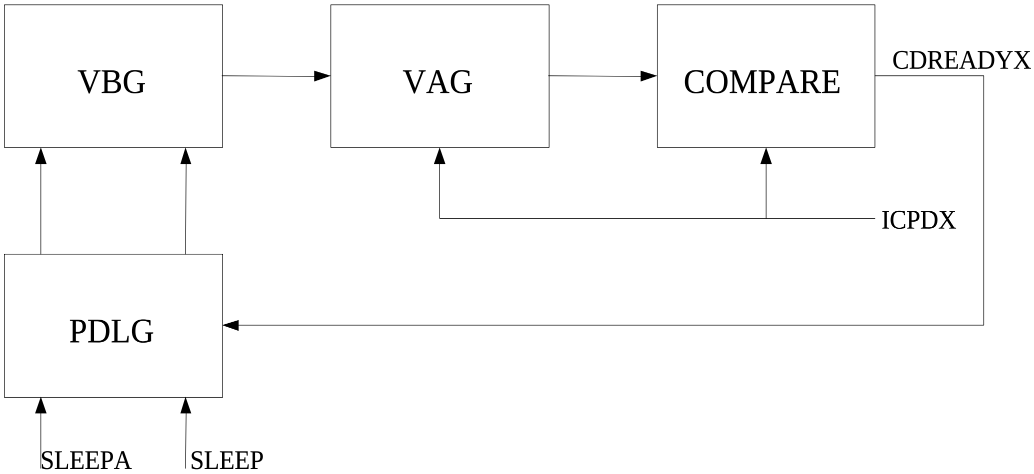
.Power Down Mode

The CODEC will recover from power-down mode when **ICPDX** keeps high;

. Support system power (Adapter and Battery) detection. The function will work well even under 3V power Supply;

. Support power-down control when **ICPDX** keeps low;

. Support 4 power-down modes for special applications:



MODE / FUNCTION	REG 6 (7,6) (SLEEPA,SLEEP) = (0,0)	REG 6 (7,6) (SLEEPA,SLEEP) = (0,1)	REG 6 (7,6) (SLEEPA,SLEEP) = (1,0)	REG 6 (7,6) (SLEEPA,SLEEP) = (1,1)
VBG reference	on	off	off	on
POW & ***BAT***	on	off	on	on
all analog blocks	off	off	off	on
A/D and D/A	off	off	off	off

Table 1

* Power down procedure

1. Keep (SLEEPA, SLEEP) = (0,0) in stand by mode.
2. Setup (SLEEPA, SLEEP) to system required mode.
3. Trigger CODEC power down. Clear **CDCMR ICPDX** (bit 2) = 0.
4. Trigger DSP power down. Set **CTLR PWDN** bit (bit 11) = 1.

* Wake up procedure

1. Trigger DSP wake up. Clear **CTLR PWDN** bit (bit 11) = 0.
2. Wait DSP stable. Wait **CTLR PWDNS** bit (bit 5) = 0.
3. Setup (SLEEPA, SLEEP) = (0,0).
4. Trigger CODEC wake up. Set **CDCMR ICPDX** (bit 2) = 1.
5. Wait CODEC ready. Wait **CDCMR CDREADYX** (bit 9) = 0.

.3-Channel Input (MIC,AUX1,LIN) with PRE-PGA (Pre-Programmable Gain Control)

. Input Range : 0 ~ AVDD-2Vpp;
. ***PRE-PGA*** gain step from 21dB to -15dB (21, 18, 15, 12, 9, 7.5, 6, 4.5, 3, 0, -3, -6, -9, -12, -15dB);
. Driving Capacity : more than 400uA at **FILT** and **AUX2** output;
. Input Impedance : more than 25KΩ;
. THD : less than 70dB at **FILT** output;
. There is just one path which can be selected at the same time;
. The gain setting of the path will be mapped to the ***PRE-PGA*** when user changes the path of Input.

. ALC (Automatic Level Control)

. Input Range : 0 ~ 1.2Vpp (Loop Gain : 30dB);
. Output Characteristic : see **FIG. 5** ~ **FIG. 7**;
. Loop Gain : 37dB max (with external RC time constant);
. Driving Capacity : more than 400uA at **FILT** and **AUX2** output;
. THD : less than 40dB at **FILT** output (Loop Gain : 40dB).

. AD1 PGA

. Input Range : 0 ~ AVDD-2Vpp;
. ***AD1-PGA*** can support gain step from 0dB to 18dB (0, 4, 8, 18dB);

. AD2 PGA

. Input Range : 0 ~ AVDD-2Vpp;
. ***AD2-PGA*** can support gain step from -6dB to 39dB (-6, -3, 0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39dB);

. FILT as I/O Port

. Input Range : 0 ~ AVDD-2Vpp;
. Input Impedance : more than 1KΩ;
. Output Impedance : less than 1KΩ;
. Load Capacitance : 5000pF;

. AUX1 & AUX2 as I/O Port

. Input Range : 0 ~ AVDD-2Vpp;
. Input Impedance : more than 15KΩ;
. Output Impedance : less than 15KΩ;

. External passive L.P.F. (Low Pass Filter)

. External capacitors (**LPFC1** and **LPFC2**) can be changed to attenuate high frequency noise at **SPKP** and **SPKN** output;
. When external capacitors (**LPFC1** and **LPFC2**) are NC (no connection), then passive ***L.P.F.*** will be by-passed;

. Output of the Line Driver (**LOUTP** and **LOUTN**) can be chosen to pass or by-pass the ***L.P.F.***;
. **LPFC1/LPFC2** can be a D/A output pin and output impedance is around 3KΩ/6KΩ;

. D/A PGA

. Input Range : 0 ~ AVDD-2Vpp;
. ***DA-PGA*** can support gain step from 0dB to 6dB (2dB/step);

. Line Driver (*LIN-DRV*)

. Not only support the programmable gain from 0 to 22.5dB, but also fully differentially drive 6Vpp over 600Ω;
. If switches ***SWE***, ***SWJ***, ***SWK*** and ***SWL*** are opened, then the line driver will be muted to -70dB and power-down automatically;

1. output swing : Single Ended (only use **LOUTP** or **LOUTN**) : 0 ~ 3Vpp (over 600Ω load, at LIN-DRV = 0dB); Fully differential (use **LOUTP** + **LOUTN**) : 0 ~ 6Vpp (over 600Ω load, at LIN-DRV = 0dB);
2. ***LIN-DRV*** gain step from 0dB to 22.5dB (1.5dB/step);
3. THD : less than 70dB at 6Vpp output over 600Ω load;

. Attenuator (*ATT1* & *ATT2*)

. Speaker output signal can be attenuated either by internal register or external resistor;
. If switches ***SWF*** and ***SWH*** are opened, then attenuator will be muted to -70dB automatically;

1. ***ATT1*** (internal register) : 16 steps programmable, from -45dB to 0dB (-45, -39, -33, -27, -24, -21, -18, -15, -12, -9, -7.5, -6, -4.5, -3, -1.5, 0dB);
2. ***ATT2*** (external variable resistor) : from -45 ~ 0dB (determined by external 10KΩ potentiometer);
3. THD : less than 70dB;
4. input range for **AUX2** : 0 ~ AVDD-2Vpp;
5. input impedance for **AUX2** : more than 15KΩ;

. Speaker Driver (SPK-DRV)

. If switches ***SWF*** and ***SWH*** are opened, then ***SPK-DRV*** will be power-down automatically;

1. Maximum output swing : 6Vpp with 8Ω load at fully differential output (**SPKP** + **SPKN**);
2. THD : less than 60dB (at 6Vpp/8Ω load);

. Voltage Reference (VREF & VAG)

. Two 2.25V± voltage references are on-chip generated, where **VREF** is for external circuit use and **VAG** is for internal circuit use;
. **VREF** can be used to bias the microphone, the level shift circuit or other applications;

1. **VREF** driving capacity : more than 400uA;
2. User can use the **VREF** to provide DC bias to external components;

. Bandgap Reference (VBG)

. A bandgap circuit generates a voltage source (**VBG**) which is around 1.2V⊕.It is with low temperature coefficient and good power supply rejection;

. If user changes VBG bypass capacitor (C15) then the MX93002 warm-up time will be changed; see **The Timing Diagram of CODEC Function;**

. Serial Control Interface

. Use **IFS** for synchronization with **ISDATAW/ISDATAR** to read/write the internal control registers;

. All registers will keep original setting when the CODECs returns from power-down or sleep mode;

1. When **ISDENX** (serial data enable) signal active low, the CODECs starts to receive(transmit) serial control data **ISDATAW (ISDATAR)**;
2. Set **ISDENX** from low to high when transmitting / receiving **ISDATAW / ISDATAR** is complete;
3. **ISDATA(R/w)** format : 3 addresses from A2 to A0, 8 data from D7 to D0 (A2 is MSB and D0 is LSB);

. Two Comparators for AC power and battery power

. To detect AC power and battery power or other applications;

1. input range : 0 ~ AVDD-2Vpp (with 7V surge protection);
2. input impedance : more than 10^12Ω ;
3. input offset voltage : less than 10mV;
4. output impedance : less than 10KΩ;
5. slew rate : 3V/us max.;

. Switches

. There are three registers (REG0, REG3 and REG6) which are used to control all of the switches so that user can direct many different signal paths, for examples:

1. Record signal from **MIC** and play signal to **SPKP/N** or play signal to **LOUTP/N**:

   A. Record signal from **MIC** or Record signal from **LIN**:

   a. System initialization [set **MIC** gain (REG2 bit(3~0)), set **LIN** gain (REG1 bit(7~4), set ***ALC*** gain 0/6dB (REG5 bit(1)) and set ***A/D-PGA*** gain (REG6 bit(1,0))]

   b. Record signal from **MIC** : set REG0 = 0X0048
   MIC ⇒ ***SWA*** ⇒ ***PRE-PGA*** ⇒ ***SWC*** (***ALC*** on) ⇒ ***SWD*** ⇒ ***AD1-PGA*** ⇒ ***PCM CODEC AIN1***

   c. Record signal from **LIN** : set REG0 = 0X00C8
   LIN ⇒ ***SWA*** ⇒ ***PRE-PGA*** ⇒ ***SWC*** (***ALC*** on) ⇒ ***SWD*** ⇒ ***AD1-PGA*** ⇒ ***PCM CODEC AIN1***

   B. Play signal to **SPKP/N** or play signal to **LOUTP/N**:

   a. System initialization [fix the value of ***L.P.F.*** , set (REG6 bit(5)), set ***D/A-PGA*** gain (REG6 bit(3,2), set ***ATT1*** gain (REG3 bit(3~0)) and ***LIN-DRV*** gain (REG1 bit(3~0))]

   b. Play signal to **SPKP/N** (use digital volume control) : set REG 0 = 0X0003
   PCM CODEC AOUT1 ⇒ ***L.P.F.*** ⇒ ***SWF*** ⇒ ***DA-PGA*** ⇒ ***SWG*** (***ATT1***) ⇒ ***SPK-DRV*** ⇒ **SPKP/N**

c. Play signal to **LOUTP/N** : set REG 0 = 0X0004

i. ***PCM CODEC AOUT1*** ⇒ ***L.P.F.*** ⇒ ***SWL*** ⇒ ***LIN-DRV*** ⇒ **LOUTP/N**

ii. ***PCM CODEC AOUT2*** ⇒ ***SWE*** ⇒ ***LIN-DRV*** ⇒ **LOUTP/N**

d. Play signal to **SPKP/N** (use digital volume control) and **LOUTP/N** : set REG 0 = 0X0007

i. ***PCM CODEC AOUT1*** ⇒ ***L.P.F.*** ⇒ ***SWF*** ⇒ ***DA-PGA*** ⇒ ***SWG*** (***ATT1***) ⇒ ***SPK-DRV*** ⇒ **SPKP/N**

PCM CODEC AOUT2 ⇒ ***SWE*** ⇒ ***LIN-DRV*** ⇒ **LOUTP/N**

ii. ***PCM CODEC AOUT1*** ⇒ ***L.P.F.*** ⇒ ***SWF*** ⇒ ***DA-PGA*** ⇒ ***SWG*** (***ATT1***) ⇒ ***SPK-DRV*** ⇒ **SPKP/N**

⇒ ***SWL*** ⇒ ***LIN-DRV*** ⇒ **LOUTP/N**

2. Room Monitoring:

A. System initialization [set **MIC** gain (REG2 bit(3~0)), set ***ALC*** gain 0/+6dB (REG5 bit(1)), set **LIN-DRV** gain (REG1 bit(3~0)), set REG3 bit(6,5) and set REG6 bit(1,0)]

B. Switches path:

a. Remote Monitoring:

MIC ⇒ ***SWA*** ⇒ ***PRE-PGA*** ⇒ ***SWC*** (***ALC*** on) ⇒ ***SWJ*** ⇒ ***LIN-DRV*** ⇒ **LOUTP/N**

b. Local Detecting DTMF:

LIN ⇒ ***SWI*** ⇒ ***AD1-PGA*** ⇒ ***PCM CODEC AIN1***

3. Digital Speakerphone:

A. System Initialization [set **MIC** gain (REG2 bit(3~0)), set ***AD1-PGA*** gain (REG6 bit(1,0)), fix the value of ***L.P.F.***, set ***DA-PGA*** gain (REG6 bit(3,2)), set ***ATT1*** gain (REG3 bit(3~0)), set **LIN** gain (REG1 bit(7~4)), set ***SWM*** REG4 bit(4), set ***LIN-DRV*** gain (REG1 bit(3~0))]

B. Switches path : set REG0 = 0X00AF

a. CODEC 1 : Record signal from **MIC** and Play signal to **SPKP/N** (use digital volume control)

MIC ⇒ ***SWA*** ⇒ ***PRE-PGA*** ⇒ ***SWC*** (***ALC*** off) ⇒ ***SWD*** ⇒ ***AD1-PGA*** ⇒ ***PCM CODEC AIN1***

PCM CODEC AOUT1 ⇒ ***L.P.F.*** ⇒ ***SWF*** ⇒ ***SWG*** (***ATT1***) ⇒ ***SPK-DRV*** ⇒ **SPKP/N**

b. CODEC 2 : Record signal from **LIN** and Play signal to **LOUTP/N**

LIN ⇒ ***SWM*** ⇒ ***AD2-PGA*** ⇒ ***PCM CODEC AIN2***

PCM CODEC AOUT2 ⇒ ***SWE*** ⇒ ***LIN-DRV*** ⇒ **LOUTP/N**

. Power Consumption of CODEC (with 600W line load and 8W speaker load)

Operation \ Max. Power Consumption	***LIN-DRV*** **Dis/Enable**	***SPK-DRV*** **Dis/Enable**	**Analog circuits**	**Unit**
Stand-by	Disable	Disable	20	mA
Operating	Disable	Disable	20	mA
	Enable	Disable	31	
	Disable	Enable	217	
	Enable	Enable	217	
Power-down	Disable	Disable	120	uA
Power-down with SLEEP = 1	Disable	Disable	20	uA

@ Test condition : 1. at ***LIN-DRV*** (with 600Ω load) / ***SPK-DRV*** (with 8Ω load) full swing output
2. see ***LIN-DRV*** and ***SPK-DRV*** Descriptions

6.3 CONTROL REGISTERS DEFINITION

REGISTER 0

ADDRESS BIT	A2	A1	A0
DATA	0	0	0

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER-ON	0	1	0	1	1	1	0	0
DESCRIPTION	*SWA*		*SWB*	*SWC*	*SWD*	*SWE*	*SWF*	*SWG*

(***SWA***) D(7,6) = (1,1) : path of ***SWA*** is "c ⇒ A", ***PRE-PGA*** setting follows **LIN** GAIN SETTING
= (1,0) : path of ***SWA*** is "b ⇒ A", ***PRE-PGA*** setting follows **AUX1** GAIN SETTING
= (0,1) : path of ***SWA*** is "a ⇒ A", ***PRE-PGA*** setting follows **MIC** GAIN SETTING
= (0,0) : path of ***SWA*** is "d ⇒ A", (GROUNDING to **AG**)
(***SWB***) D(5) = (1) : path of ***SWB*** is "CLOSE", D(5) = (0) : path of ***SWB*** is "OPEN"
(***SWC***) D(4) = (1) : path of ***SWC*** is "b ⇒ A", D(4) = (0) : path of ***SWC*** is "a ⇒ A"
(***SWD***) D(3) = (1) : path of ***SWD*** is "CLOSE", D(3) = (0) : path of ***SWD*** is "OPEN"
(***SWE***) D(2) = (1) : path of ***SWE*** is "CLOSE", D(2) = (0) : path of ***SWE*** is "OPEN";
(***SWF***) D(1) = (1) : path of ***SWF*** is "CLOSE " , D(1) = (0) : path of ***SWF*** is " OPEN "
(***SWG***) D(0) = (1) : path of ***SWG*** is "a ⇒ A", ***ATTENUATOR 1 (ATT1)***
= (0) : path of ***SWG*** is "a ⇒ B", ***ATTENUATOR 2 (ATT2)***

REGISTER 1

ADDRESS BIT	A2	A1	A0
DATA	0	0	1

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER-ON	0	0	0	0	0	0	0	0
DESCRIPTION	**LIN** GAIN SETTING (***PRE-PGA***)				***LIN-DRV*** GAIN SETTING			

(**LIN** GAIN SETTING) D(7~4) = (F) ~ (0) : 21dB ~ -15dB; see **NOTE 1**
(***LIN-DRV*** GAIN SETTING) D(3~0) = (F) ~ (0) : 22.5dB ~ 0dB 1.5dB/step; see **NOTE 4**

REGISTER 2

ADDRESS BIT	A2	A1	A0
DATA	0	1	0

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER-ON	0	0	0	0	0	1	0	1
DESCRIPTION	**AUX1** GAIN SETTING (***PRE-PGA***)				**MIC** GAIN SETTING (***PRE-PGA***)			

(**AUX1** GAIN SETTING) D(7~4) = (F) ~ (0) : 21dB ~ -15dB; see **NOTE 1**
(**MIC** GAIN SETTING) D(3~0) = (F) ~ (0) : 21dB ~ -15dB; see **NOTE 1**

REGISTER 3

ADDRESS BIT	A2	A1	A0
DATA	0	1	1

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER-ON	0	0	0	0	1	1	1	1
DESCRIPTION	***SWH***	***SWI***	***SWJ***	***SWK***	***ATT1*** GAIN SETTING			

(***SWH***) D(7) = (1) : path of ***SWH*** is "CLOSE", D(7) = (0) : path of ***SWH*** is "OPEN"
(***SWI***) D(6) = (1) : path of ***SWI*** is "CLOSE", D(6) = (0) : path of ***SWI*** is "OPEN"
(***SWJ***) D(5) = (1) : path of ***SWJ*** is "CLOSE", D(5) = (0) : path of ***SWJ*** is "OPEN"
(***SWK***) D(4) = (1) : path of ***SWK*** is "CLOSE", D(4) = (0) : path of ***SWK*** is "OPEN"
(***ATT1*** GAIN SETTING) D(3~0) = (F)~(0) : - 45dB ~ 0dB; see **NOTE 3**

REGISTER 4

ADDRESS BIT	A2	A1	A0
DATA	1	0	0

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER_ON	0	0	1	0	1	0	0	0
DESCRIPTION	***AD2-PGA*** GAIN SETTING				***SWM***			

(***AD2-PGA*** GAIN SETTING) D(7~4) = (0) ~ (F) : -6dB ~ 39dB; see **NOTE 3**
(***SWM***) D(3) = (1) : path of ***SWM*** is "CLOSE", D(3) = (0) : path of ***SWM*** is "OPEN"

REGISTER 5

ADDRESS BIT	A2	A1	A0
DATA	1	0	1

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER_ON	0	0	0	0	0	0	0	0
DESCRIPTION	ALC1	SPKHI	REV	REV	REV	REV	ALC0	REV

D (5 ~ 2 and 0) : reserved
(SPKHI) D(6) = (0) : **SPKP/N** can drive 8Ω load when ***SPK-DRV*** turns on
D(6) = (1) : **SPKP/N** appears high impedance (10KΩ) and ***SPK-DRV*** will keep a quiescent current when ***SPK-DRV*** turns on(ALC1 , ALC0) D(7,1) = (0,0) : ***ALC*** open loop gain is 38dB
(ALC1,ALC0) = (0,1) : ***ALC*** open loop gain is 36dB
= (1,0) : reserved
= (1,1) : external ***ALC*** option (***PRE-PGA*** Output : **ALCC1**, ***SWC*** path “a” Input : **ALCC2**)

REGISTER 6

ADDRESS BIT	A2	A1	A0
DATA	1	1	0

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER_ON	0	0	0	0	0	0	0	0
DESCRIPTION	SLEEPA	SLEEP	***SWL***	SPK-MUTE	***SPK-DRV*** GAIN SETTING		***AD1-PGA*** GAIN SETTING	

(SLEEPA , SLEEP) D(7,6) = (0,0) : when the CODEC gets into power down mode, all the blocks of the CODEC will be disabled except the **VBG** reference and 2 comparators (***POW***, ***BAT***)

D(7,6) = (0,1) : when the CODEC gets into power down mode, all the blocks of the CODEC will be disabled

D(7,6) = (1,0) : when the CODEC gets into power down mode, all the blocks of the CODEC will be disabled except 2 comparators (***POW***, ***BAT***)

D(7,6) = (1,1) : when the CODEC gets into power down mode, all the analog blocks of the CODEC will be still functional and can be programmed by control registers

(***SWL***) D(5) = (1) : path of ***SWL*** is "CLOSE", D(5) = (0) : path of ***SWL*** is "OPEN"; see **NOTE 7**

(SPK-MUTE) D(4) = 1 : force ***SPK-DRV*** mute to -70dB, D(4) = 0 : force ***SPK-DRV*** un-mute

(***SPK-DRV*** GAIN SETTING) D(3,2) = (0,0) ~ (1,1) : 0dB ~ 8dB; 2dB/step; see **NOTE 5**

(***AD1-PGA*** GAIN SETTING) D(1,0) = (0,0) ~ (1,1) : 0dB ~ 18dB; see **NOTE 2**

REGISTER 7

ADDRESS BIT	A2	A1	A0
DATA	1	1	1

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
POWER_ON	0	0	0	0	0	0	0	0
DESCRIPTIN			***SWO***	***SWN***	READ	REGISTER ADDRESS		

(***SWO***) D(5) = (1) : path of ***SWO*** is "CLOSE", D(5) = (0) : path of ***SWO*** is "OPEN"

(***SWN***) D(4) = (1) : path of ***SWN*** is "CLOSE", D(5) = (0) : path of ***SWN*** is "OPEN"; see **NOTE 7**

(READ) D(3) = 1 : read data from Register 0 ~ 7, D(3) = 0 : write data to Register 0 ~ 7

(REGISTER ADDRESS) D(2~0) : When READ = 1, then

a. READ will be cleared automatically;

b. if next DSP **ISDENX** signal active low, the content of REGISTER ADDRESS will be dumped out through CODEC **ISDATAR** interface;

NOTE 1 : ***PRE-PGA*** gain step; from -15dB to 22dB

1111	1110	1101	1100	1011	1010	1001	1000
21dB	18dB	15dB	12dB	9dB	7.5dB	6dB	4.5dB

0111	0110	0101	0100	0011	0010	0001	0000
3.0dB	1.5dB	0dB	-3dB	-6dB	-9dB	-12dB	-15dB

NOTE 2 : ***AD1-PGA*** gain step; from 0dB to 18dB

00	01	10	11
0dB	4dB	8dB	18dB

NOTE 3 : ***AD2-PGA*** gain step; from -6dB to 39dB; 3dB/step

1111	1110	1101	1100	1011	1010	1001	1000
39dB	36dB	33dB	30dB	27dB	24dB	21dB	18dB

0111	0110	0101	0100	0011	0010	0001	0000
15dB	12dB	9dB	6dB	3dB	0dB	-3dB	-6dB

NOTE 4 : ***LIN-DRV*** gain step; from 0dB to 22.5dB; 1.5dB/step

1111	1110	1101	1100	1011	1010	1001	1000
22.5dB	21dB	19.5dB	18dB	16.5dB	15dB	13.5dB	12dB

0111	0110	0101	0100	0011	0010	0001	0000
10.5dB	9dB	7.5dB	6dB	4.5dB	3dB	1.5dB	0dB

NOTE 5 : ***SPK-DRV*** gain step; from 0dB to 6dB; 2dB/step

00	01	10	11
0dB	2dB	4dB	6dB

NOTE 6 : ***ATT1*** (Attenuator 1) gain step; from 0dB to -45dB

1111	1110	1101	1100	1011	1010	1001	1000
-45 dB	-39 dB	-33 dB	-27 dB	-24 dB	-21 dB	-18 dB	-15 dB

0111	0110	0101	0100	0011	0010	0001	0000
-12 dB	-9 dB	-7.5 dB	-6 dB	-4.5 dB	-3 dB	-1.5 dB	0 dB

NOTE 7 : 1. ***SWE***, ***SWJ*** and ***SWL*** can not be turned on at the same time;

2. ***SWJ*** and ***SWN*** can not be turned on at the same time;
3. If ***SWE***, ***SWJ*** or ***SWL*** is turned on, then ***SWK*** will be taken as an output port;
4. If ***SWK*** is taken as an input port, ***SWK***, ***SWE***, ***SWJ*** and ***SWL*** cannot be turned on at the same time;

7.1 DC CHARACTERISTICS:

SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
	Operation temperature		0		70	^{0}C
	Storage temperature		-55		150	^{0}C
	Operation Frequency			40.96		MHz
VCC	Supply Voltage		4.5	5	5.5	Volt
GND	Ground			0		Volt
VIH	Input high voltage	Schmidt trigger input(IS)	0.7*VCC			Volt
VIL	Input low voltage	Schmidt trigger input(IS)			0.3*VCC	Volt
RH	Pull high register	for IPT[3:0] pins		150 K		ohm
IOL(OA)	Output low current	@VOL=0.4	8			mA
IOL(OB)	Output low current	@VOL=0.4	16			mA

Absolute Maximum Rating

PARAMETER	MIN	TYP	MAX	UNITS
AVDD to AGND	-0.3		6.0	V
VDD to DGND	-0.3		6.0	V
Voltage at any Digital Input or Output	DGND-0.3		VDD+0.3	V
Current at any Digital Input or Output			8	mA
Operating Ambient Temperature Range	0		70	¢ J
Storage Temperature Range	-65		150	¢ J
Lead Temperature (Soldering, 10 seconds)			260	¢ J

Power Supply

PARAMETER	MIN	TYP	MAX	UNITS
Power Supply Voltage :				
Digital and Analog	4.5	5.0	5.5	V
Power Supply Current :				
Stand-by :				
Digital		10	12	mA
Analog		20		mA
Operating :				
Digital		60	70	mA
Analog (see Page 77)				mA
Power-Down :				
Digital			2	mA
Analog (at REG4 bit 6 SLEEP = 0)			120	uA
Analog (at REG4 bit 6 SLEEP = 1)			20	uA

Electrical Characteristics (**BOLD** characters are guaranteed for AVDD = VDD = 5V ±5%, Temperature = 0 ~ 70 ¢ J Typical specified at AVDD = VDD = 5V, temperature = 25 ¢ J "*****" mark : guaranteed by design)

Analog Input Ports

PARAMETER	MIN	TYP	MAX	UNITS
MIC / LIN / AUX1 :				
Input Voltage			3.0	Vpp
* Input Capacitance			15	pF
* Input Impedance	20			KΩ

Analog Output Ports

PARAMETER	MIN	TYP	MAX	UNITS
Line Driver :				
Gain Range	0		22.5	dB
Step Variation		0.3		dB
Fully Differential (LOUTP+LOUTN) Full Swing / with 600Ω load		6.0		Vpp
Single Ended (LOUTP) Full Swing / with 600Ω load		3.0		Vpp
* External Load Capacitance			200	pF
* Output Loading	600			Ω
Speaker Driver :				
Fully Differential (SPKP+SPKN) Full Swing / with 8Ω load		6.0		Vpp
Single Ended (SPKP) Full Swing / with 8Ω load		3.0		Vpp
* External Load Capacitance			100	pF
* Output Loading	8			Ω
the Quiescent current (when REG5 bit(6) SPKHI = 1)			4	mA

Analog I/O Ports

PARAMETER	MIN	TYP	MAX	UNITS
FILT :				
as Input Port :				
* Input Capacitance	5000			pF
* Input Impedance	1			KΩ
as Output Port :				
* External Load Capacitance			5000	pF
* Output Impedance			1	KΩ
AUX2 :				
as Input Port :				
* Input Capacitance	15			pF
* Input Impedance	15			KΩ
as Output Port :				
* External Load Capacitance			15	pF
* Output Impedance			15	KΩ

Gain Variation

PARAMETER	MIN	TYP	MAX	UNITS
PRE-PGA :				
Gain Range	-15		22.5	dB
Step Size		±1.5, ±3		dB
Step Variation		±0.3		dB
AD-PGA :				
Gain Range	0		9	dB
Step Size		+3		dB
Step Variation		±0.3		dB
DA-PGA :				
Gain Range	0		9	dB
Step Size		+3		dB
Step Variation		±0.3		dB

Attenuator

PARAMETER	MIN	TYP	MAX	UNITS
Attenuator 1 (Digital Volume) :				
Gain Range	-45		0	dB
Step Size		-6, -3, -1.5		dB
Step Variation		±0.3		dB
* Mute Attenuation		-70		dB
Attenuator 2 (External Volume) :				
Gain Range	-45		0	dB
the Requirement of External Resistor (from SPKP to VR)		10		KΩ
* Mute Attenuation		-70		dB

Voltage Reference (VREF pin)

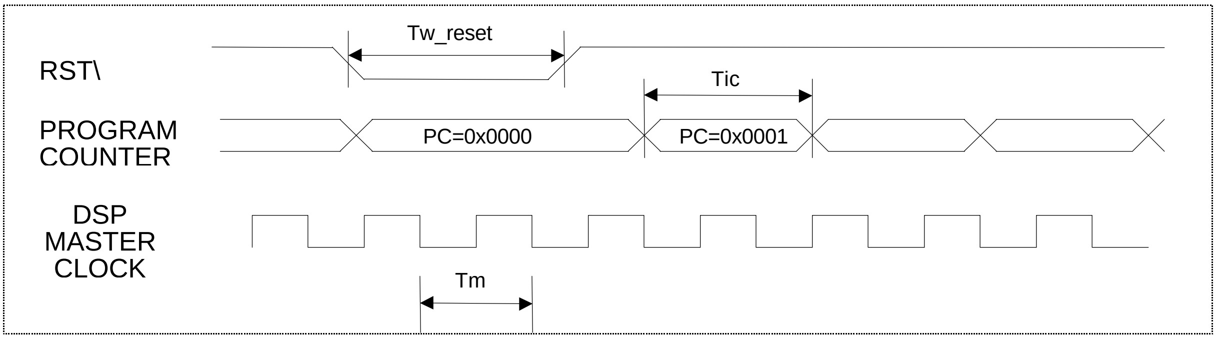
PARAMETER	MIN	TYP	MAX	UNITS
Output Voltage	2.0	2.25	2.5	V
* Output Current		450		uA

Two Comparators (POW, BAT)

PARAMETER	MIN	TYP	MAX	UNITS
Input Voltage (VCOMP, VPOW, VBAT)			AVDD	V
* Hysteresis		15		mV
* Output Impedance of POWB and BATB pins	10			KΩ

7.2 AC TIMING and CHARACTERISTICS:

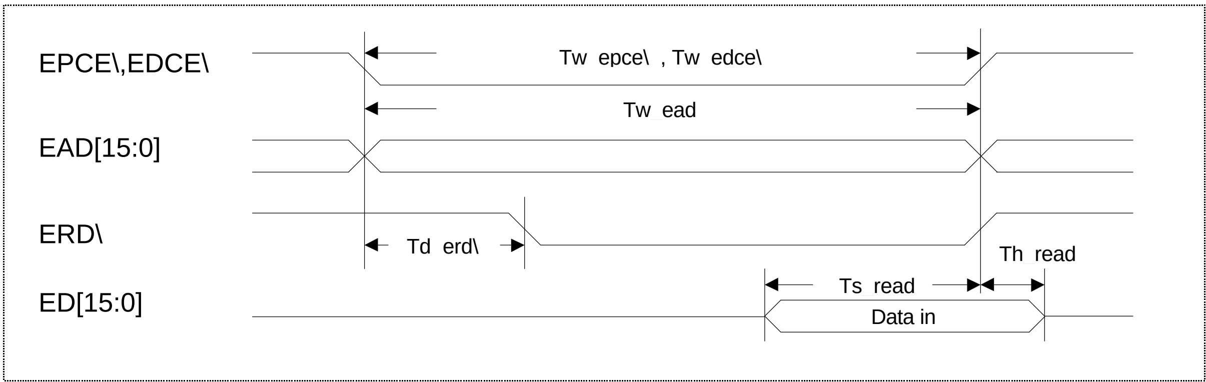
7.2.1 RESET TIMING



SYMBOL	PARAMETER	CONDITION	MIN TYP MAX	UNIT
Tm	Master clock cycle time	@ 40.96 MHz	24.4	ns
Tic	Instruction cycle time	@ 40.96 MHz 0 wait state	36.6	ns
Tw_reset	Reset low pulse width		3*Tm	ns

NOTE : PLL output clock will be reset to a lower frequency (around 24 ~ 25 MHz) during power on reset or at the starting point when DSP just comes back from power down mode. It takes about 10 ms for FLL to lock at the target frequency specified in **PLLMR**(I/O mapped 21) when RST\ pin going high or **PWDN** bit being cleared.

7.2.2 EXTERNAL PROGRAM and DATA READ TIMING

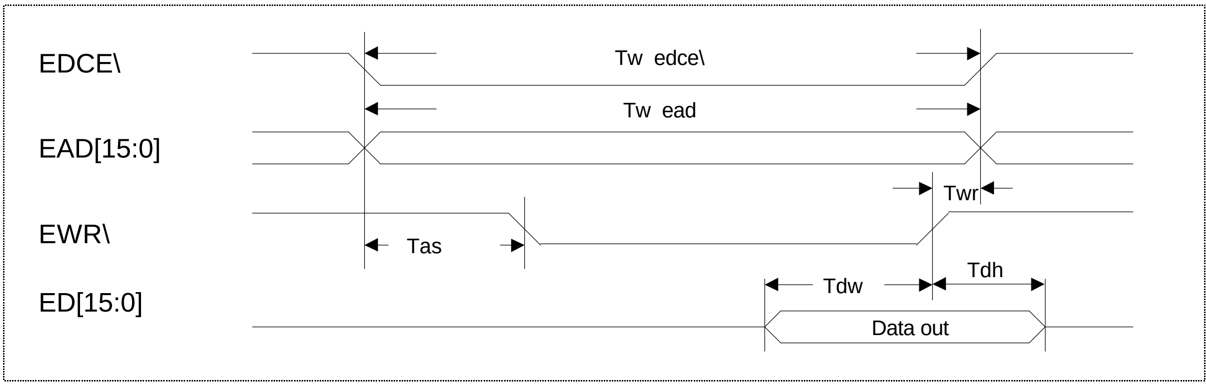


SYMBOL	PARAMETER	CONDITION	MIN TYP MAX	UNIT
Tw_epce\	Program read cycle time	see Note1	Tm*(1.5+Wp)	ns
Tw_edce\	Data read cycle time	see Note1	Tm*(1.5+Wd)	ns
Tw_ead	Read address cycle time		Same as Tw_epce\ and Tw_edce\	ns
Td_erd\	Read enable delay time		Tm*0.5	ns
Ts_read	Data read setup time	see Note2	20 or 40	ns
Th_read	Data read hold time		0 10 15	ns

NOTE1: Wp is **PROGWAIT[2:0]** in **WSTR ,** Wd is **DATAWAIT[2:0]** in **WSTR**

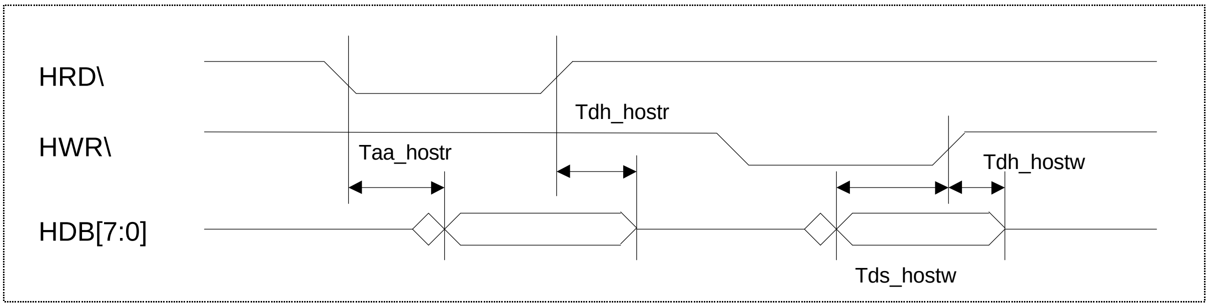
NOTE2: Ts_read : 20 ns when **FAST** (in **EXTCTLR**) =1 , 40 ns when **FAST** (in **EXTCTLR**) =0

7.2.3 EXTERNAL DATA WRITE TIMING



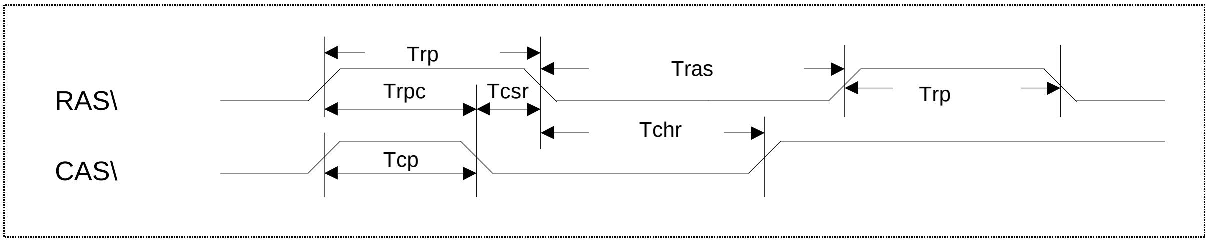
SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
Tas	Address set-up time			Tm*0.5		ns
Twr	Write recovery time		0			ns
Tdw	Data set-up time		10			ns
Tdh	Data hold time		0			ns

7.2.4 HOST INTERFACE TIMING



SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
Taa_hostr	Host read access time				50	ns
Tdh_hostr	Host read data hold time		5			ns
Tds_hostw	Data setup time at host write		40			ns
Tdh_hostw	Data hold time at host write		10			ns

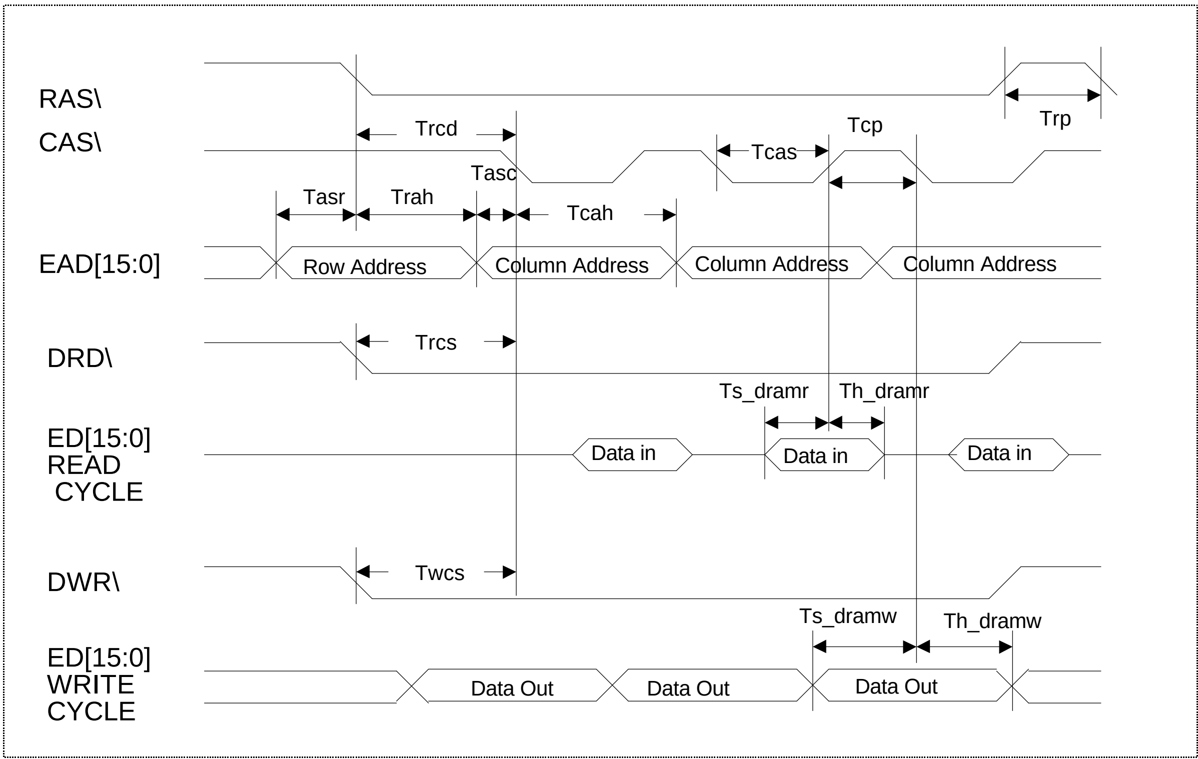
7.2.5 DRAM CAS BEFORE RAS REFRESH TIMING



SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
Trp	RAS\ precharge time	@40.96 MHz	61			ns
Trpc	RAS\ to CAS\ precharge time	“	48.8			ns
Tcsr	CAS\ setup time	“		12.2		ns
Tras	RAS\ pulse width	“		85.4		ns
Tcp	CAS\ precharge time	“	24.4			ns
Tchr	CAS\ hold time	“		48.8		ns
T_refresh	Refresh cycle time	see NOTE		15.258		us

NOTE : DSP will generate CAS\ before RAS\ self refresh every 15.258 us(32768Hzx2) .

7.2.6 DRAM READ/WRITE TIMING



SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
Trp	RAS\ precharge time		61			ns
Tcp	CAS\ precharge time			24.4		ns
Tcas	CAS\ low pulse width	see Note1		Tm*(2+W)/2		ns
Trcd	RAS\ to CAS\ delay time			48.8		ns
Tasr	Row address set-up time		0			ns
Trah	Row address hold time			24.4		ns
Tasc	Column address set-up time		0			ns
Tcah	Column address hold time		24.4			ns
Trcs	Read command set-up time		0			ns
Ts_dramr	DRAM read data set-up time	see Note2	20 or 40			ns
Th_dramr	DRAM read data hold time		0			ns
Twcs	Write command set-up time		0			ns
Ts_dramw	DRAM write data set-up time		0			ns
Th_dramw	DRAM write data hold time		36.6			ns

NOTE1: W is **DRAMWAIT[2:0]** in **WSTR**

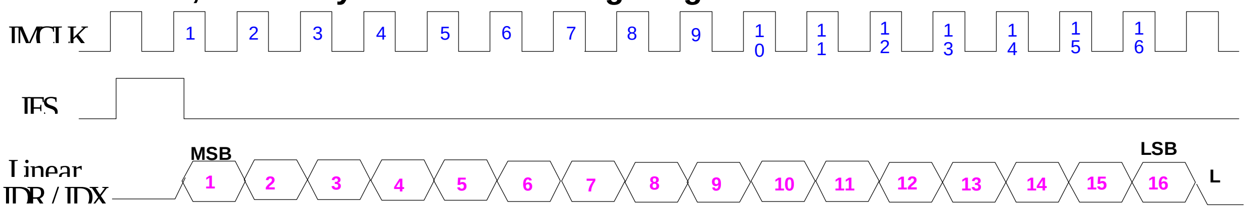
NOTE2: Ts_dramr : 20 ns when **FAST** (in **EXTCTLR**) =1
40 ns when **FAST** (in **EXTCTLR**) = 0

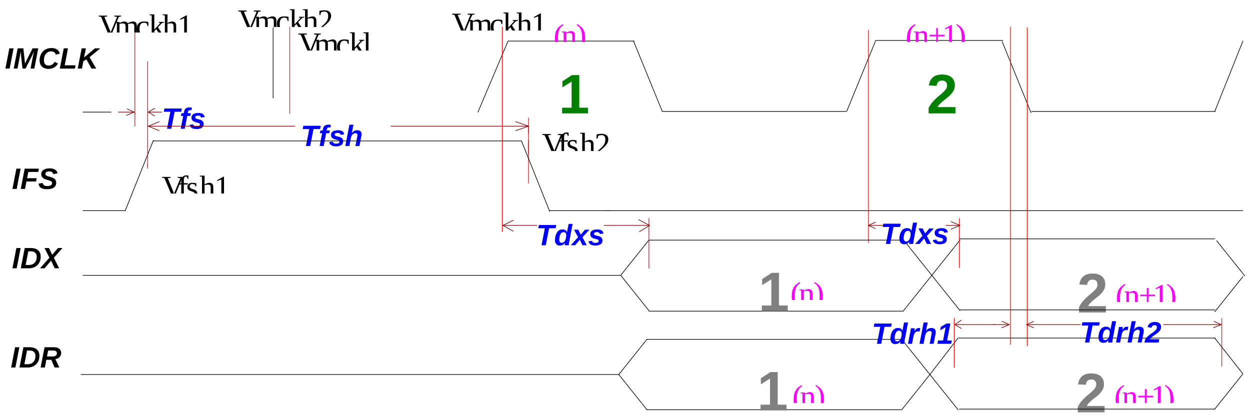
7.2.7 CODEC TIMING DESCRIPTION

TIMING	DESCRIPTION	MIN	TYP	MAX	UNIT
1/Tmck	frequency of master clock (from Vmckh1 to next Vmckh1) at RATE = 0	1.638	2.048	2.560	MHz
Trmck	rise time of master clock			50	ns
Tfmck	fall time of master clock			50	ns
Tfs	from Vmckh1 to Vfsh1	0			ns
Tfsh	holding time for frame sync. From Vfsh1 to Vfsh2	MCLK			ns
Tdxs	setting time for CODEC transmit data from Vmckh1(n) to IDX(n) data ready	110			ns
Tdrh1	holding time for CODEC received data from IDR(n) data ready to Vmckh2(n)	0			ns
Tdrh2	holding time for CODEC received data from Vmckl(n) to DR(n) ending	150			ns
Tupen1	from Vsclkh1 to Venl	40		IFS	ns
Tupen2	from Vsclkh1 to Venh	40		IFS	ns
Tups1	setting time for DSP transmitting ISDATAW from Vupenl to DSP ISDATAW(n) ready (@ where Tupen1+Tups1 must < IFS)	40		IFS	ns
Tups2	setting time for DSP transmitting ISDATAW from Vsclkh1(n+1) to DSP SDATA(n+1) ready	40		IFS	ns
Tuph	holding time for DSP transmitting ISDATAW from Vsclkh1(n+1) to DSP ISDATAW(n) ending	40		Tups 2	ns
Tcdrd	from Vsclkh1(n+1) to CODEC reading ISDATAW(n)			20	ns
Tcds1	setting time for CODEC transmitting ISDATAR from Vcdi2o to ISDATAR(n) ready			20	ns
Tcds2	setting time for CODEC transmitting ISDATAR from Vsclkh1(n+2) to ISDATAR(n+1) ready			20	ns
Tcdh	holding time for CODEC transmitting ISDATAR from ISDATAR(n) ready to Vsclkh1(n+2)			IFS	ns
Tcdo2i	from Venh to CODEC changing its ISDATAR interface to input port			20	ns
Tuprd	from Vsclkh1(n+1) to DSP reading ISDATAR(n)	40		IFS	ns
Tupi2o	from Vsclkh1 to DSP changing its ISDATAR interface to output port	40		IFS	ns
Vmckh1	logic high when CODEC IMCLK rising				
Vmckh2	logic high when CODEC IMCLK falling				
Vmckl	logic low when CODEC IMCLK falling				
Vfsh1	logic high when CODEC IFS rising				
Vsclkh1	logic high when IFS rising				
Venh	logic high when uP SDENB rising				
Venl	logic low when uP SDENB falling				

TIMING DIAGRAM

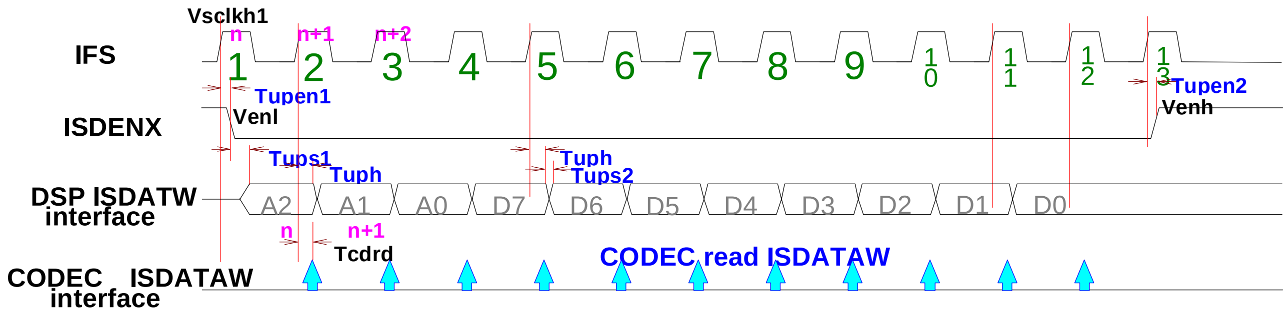
Master Clock, Frame Sync. & Data Timing Diagram



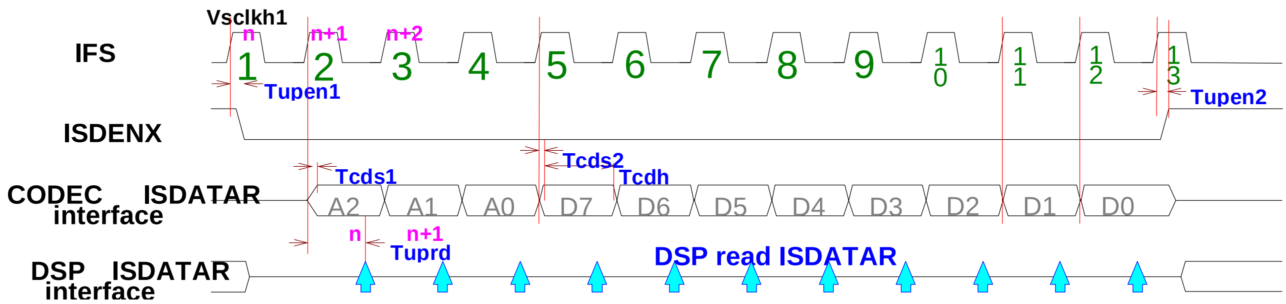


Control Registers R/W Timing Diagram

CODEC READ

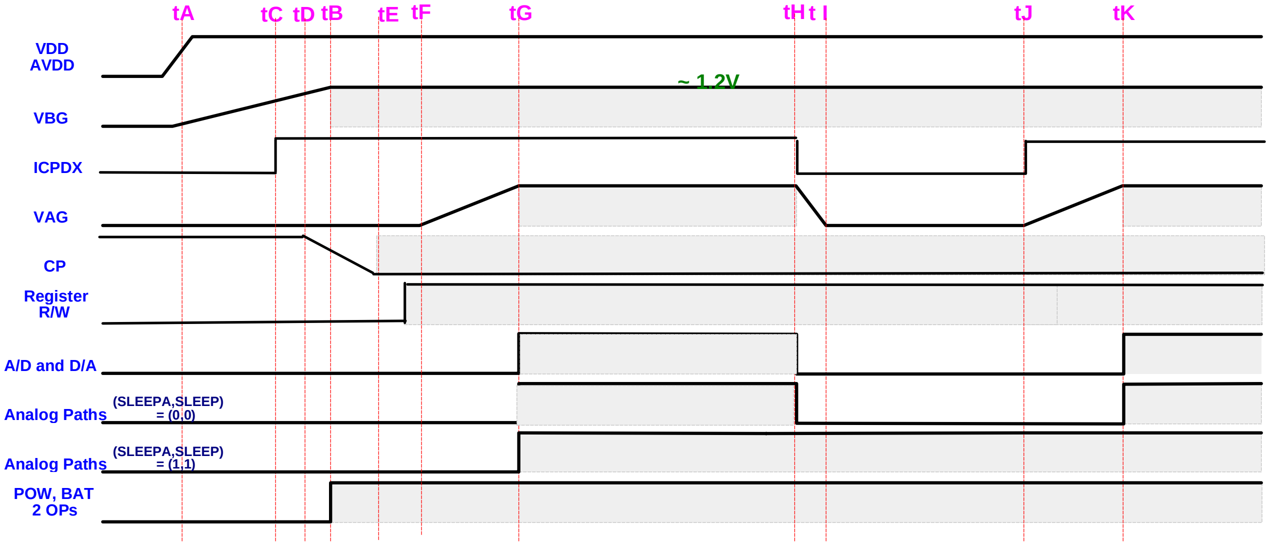


CODEC WRITE



The Timing Diagram of CODEC Function

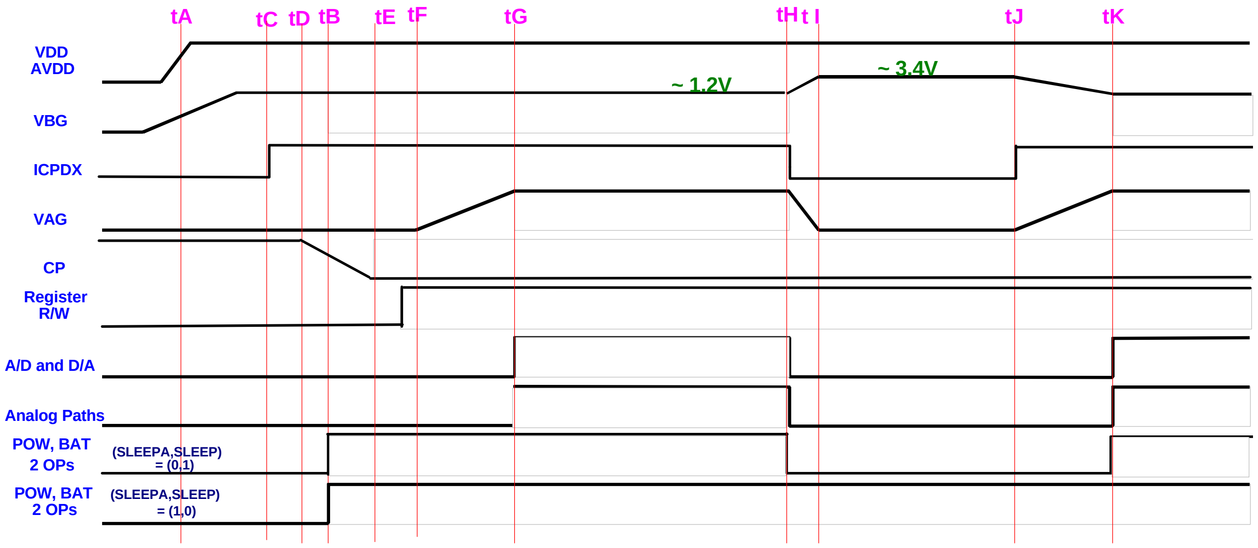
(SLEEPA,SLEEP) = (0,0) or (1,1)



@ Analog Paths : Analog I/O, Switches, PGA and Attenuator

@ [shaded box] : Stable

(SLEEPA,SLEEP) = (0,1) or (1,0)



@ Analog Paths : Analog I/O, Switches, PGA and Attenuator

@ [box] : Stable

The Timing Description of CODEC Function

TIMING	DESCRIPTION	MIN	TYP	MAX	UNIT
tA	**VDD** / **AVDD** ¡ Ü3.0VDC				
tC	**ICPDX** started				
tC ⇒ tH	**ICPDX** keeps				
tH	Power-down started (**ICPDX** keeps low)				
tJ	Power-down ended (**ICPDX** keeps high)				
tH ⇒ t J	**ICPDX** keeps low				
tA ⇒ tB	the charge time of **VBG** (where **VBG** bypass cap. = 0.1uF)	140	190	290	ms
tD ⇒ tE	the lock-in time of PLL (C1=100pF, C2=6pF, R1=68KΩ)	50	110	160	us
tF ⇒ tG **tJ ⇒ tK**	the charge time of **VAG** (where **VAG** bypass cap. = 0.1uF)	1.5	2	2.5	ms
tH ⇒ tI	the discharge time of **VAG** (where **VAG** bypass cap. = 0.1uF)	0.3	0.5	0.7	ms
th ⇒ ti	the delay time of **VBG** disable (where **VBG** bypass cap. = 0.1uF)	6	10	15	ms

@ when change **VBG** bypass capacitor (C15) :

i. from 0.1uF to 1uF : (**tA** ⇒ **tB**)' ¡ Ü10 ∗ (**tA** ⇒ **tB**)

ii. from 0.1uF to 0.01uF : (**tA** ⇒ **tB**)' ¡ Ü1/10 ∗ (**tA** ⇒ **tB**)

A/D Path Characteristics (0dBFS : reference to Fin = 1.02KHz and A/D Input is Full Swing)

PARAMETER	MIN	TYP	MAX	UNITS
Dynamic Range (at -51dBFS)	76	77	78	dB
THD+N (at Vin = -6dBFS)	-58	-62	-64	dB
Interchannel Isolation of LIN/MIC/AUX1 (at Vin =		76		dBFS
0dBFS)	-0.3		0.3	dBFS
Gain Variation (at Vin = -6dBFS)		3.0		Vpp
Max. Overload Level				
Frequency Response (Measure Respone from				
60Hz to		-23	-26	dB
4000Hz, see **FIG. 3** *) :*		-7	-8	dB
60Hz		-3	-4	dB
150Hz	-0.8		+0.8	dB
200Hz		-1.6		dB
300 ~ 3200Hz		-4.5		dB
3400Hz		-10		dB
3600Hz		-45		dB
3800Hz				
4000Hz and Up				

D/A Path Characteristics (0dBFS : reference to Fout = 1.02KHz and D/A Output is Full Swing)

PARAMETER	MIN	TYP	MAX	UNITS
Dynamic Range (at -51dBFS)	76	77	78	dB
THD+N (at Vin = -6dBFS)		46		dB
Gain Variation (at Vin = -6dBFS)		± 0.1		dBFS
Out of Band Energy (with 1.02KHz Image) :				
3.8KHz ~ 20KHz		-50		dBFS
Output Level (at AUX2)		3.0		Vpp
Frequency Response (Measure Respone from				
60Hz to				
3800Hz, see **FIG. 4** *) :*		-0.1		dB
60Hz ~ 300Hz	- 0.6		+ 0.1	dB
300Hz ~ 2800Hz		-1.1		dB
3000Hz		-2.1		dB
3200Hz		-3.7		dB
3400Hz		-6.3		dB
3600Hz		-10		dB
3800Hz				

Noise (Test Condition : 1. A/D 1 or 2 Input Signal is 1.02KHz/0dB (Full Swing)
2. D/A 1 or 2 Output Signal is 1.02KHz/0dB (Full Swing))

PARAMETER	MIN	TYP	MAX	UNITS
Idle-Channel Noise (Input Grounded and Measurement Bandwidth from 0 to 4000Hz) :				
A/D Path		-76		dB
D/A Path		-83		dB
VDD Power Supply Rejection (A/D & D/A Input Grounded and VDD = 5.0VDC+100mVrms) :				
A/D Channel : (Test Condition 1)				
Fin = 0 ~ 4KHz		-54		dB
Fin = 4 ~ 25KHz		-80		dB
Fin = 25 ~ 50KHz		-82		dB
D/A Channel : (Test Condition 2)				
Fin = 0 ~ 4KHz		-65		dB
Fin = 4 ~ 25KHz		-80		dB
Fin = 25 ~ 50KHz		-95		dB
AVDD Power Supply Rejection (A/D & D/A Input Grounded and AVDD = 5.0VDC+100mVrms) :				
A/D Channel : (Test Condition 1)				
Fin = 0 ~ 4KHz		-72		dB
Fin = 4 ~ 25KHz		-85		dB
Fin = 25 ~ 50KHz		-87		dB
D/A Channel : (Test Condition 2)				
Fin = 0 ~ 4KHz		-41		dB
Fin = 4 ~ 25KHz		-53		dB
Fin = 25 ~ 50KHz		-60		dB
Crosstalk :				
A/D 1 to A/D 2 (Test Condition 1)		-93		dB
A/D 1 to D/A 1 (Test Condition 1)		-92		dB
A/D 1 to D/A 2 (Test Condition 1)		-79		dB
A/D 2 to A/D 1 (Test Condition 1)		-98		dB
A/D 2 to D/A 1 (Test Condition 1)		-86		dB
A/D 2 to D/A 2 (Test Condition 1)		-86		dB
D/A 1 to A/D 1 (Test Condition 2)		-100		dB
D/A 1 to A/D 2 (Test Condition 2)		-94		dB
D/A 1 to D/A 2 (Test Condition 2)		-86		dB
D/A 2 to A/D 1 (Test Condition 2)		-99		dB

D/A 2 to A/D 2 (Test Condition 2)		-94		dB
D/A 2 to D/A 1 (Test Condition 2)		-86		dB

FIG. 1

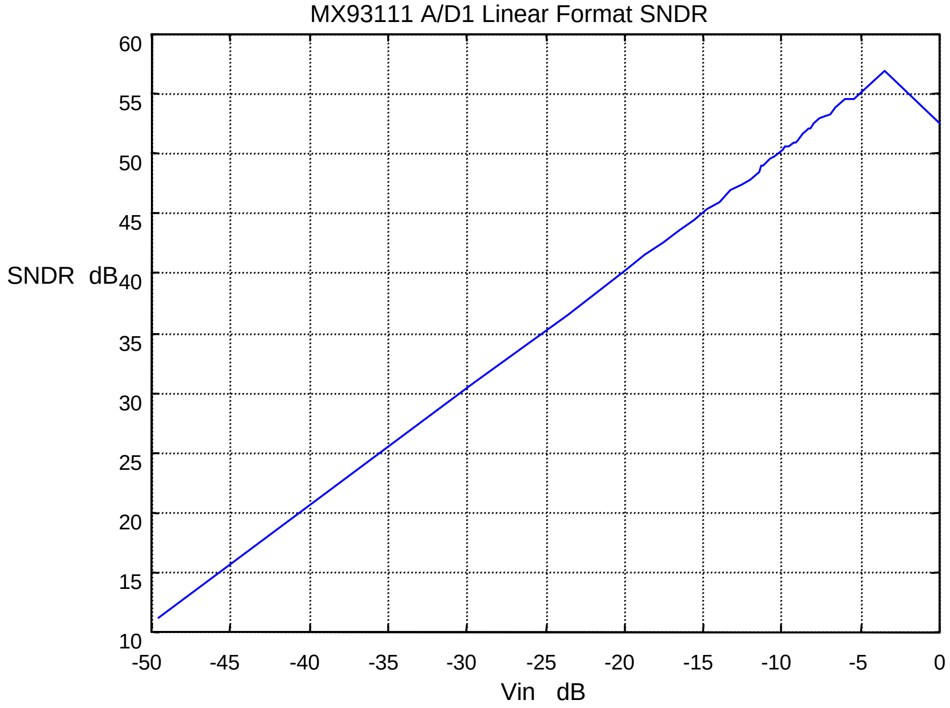

MX93111 A/D1 Linear Format SNDR
SNDR dB
60
55
50
45
40
35
30
25
20
15
10
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0
Vin dB

FIG. 2

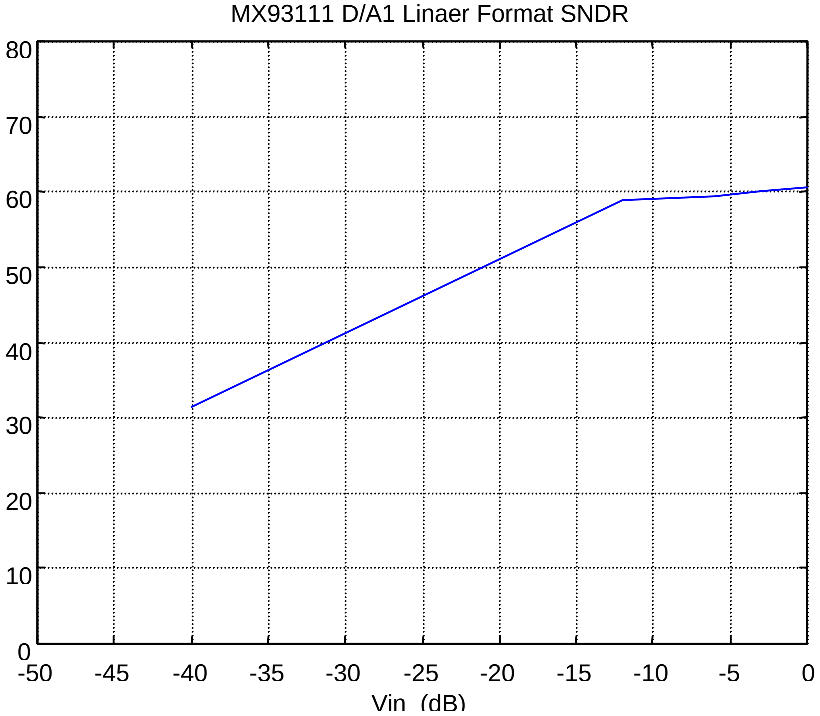

MX93111 D/A1 Linaer Format SNDR
80
70
60
50
40
30
20
10
0
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0
Vin (dB)

FIG. 3

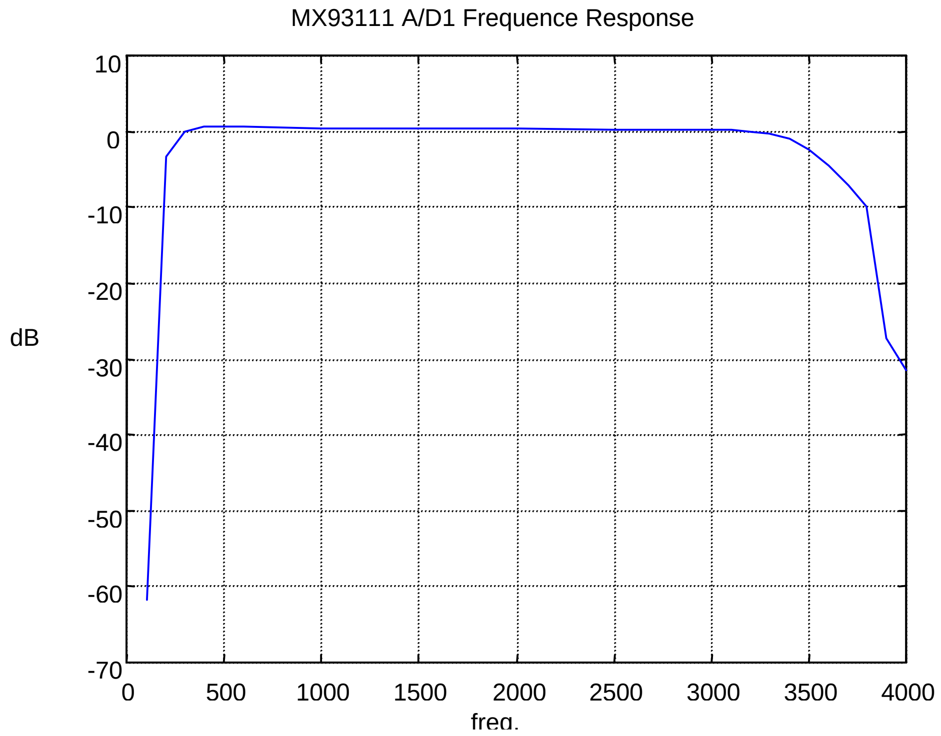

MX93111 A/D1 Frequence Response
dB
10
0
-10
-20
-30
-40
-50
-60
-70
0
500
1000
1500
2000
2500
3000
3500
4000
freq.

FIG. 4

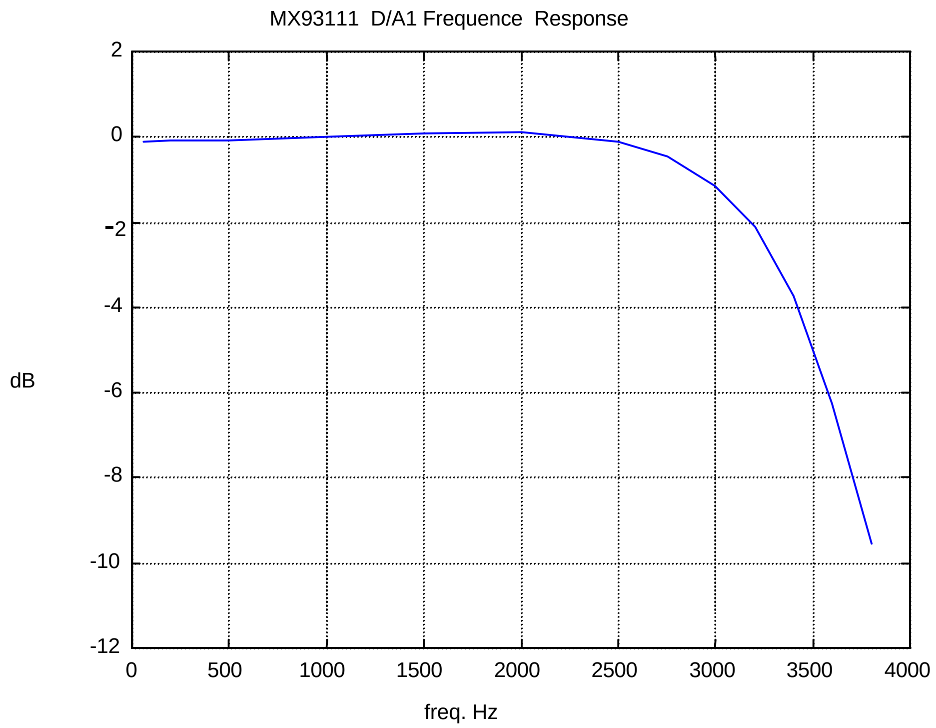

MX93111 D/A1 Frequence Response
dB
2
0
-2
-4
-6
-8
-10
-12
0
500
1000
1500
2000
2500
3000
3500
4000
freq. Hz

FIG. 5

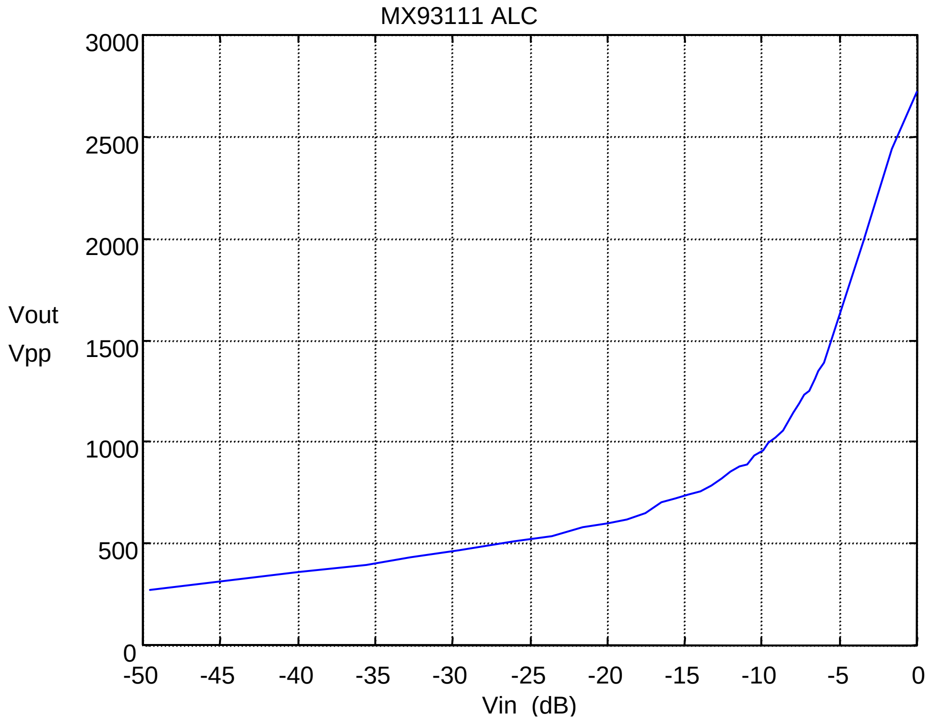

MX93111 ALC
Vout
Vpp
3000
2500
2000
1500
1000
500
0
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0
Vin (dB)

FIG. 6

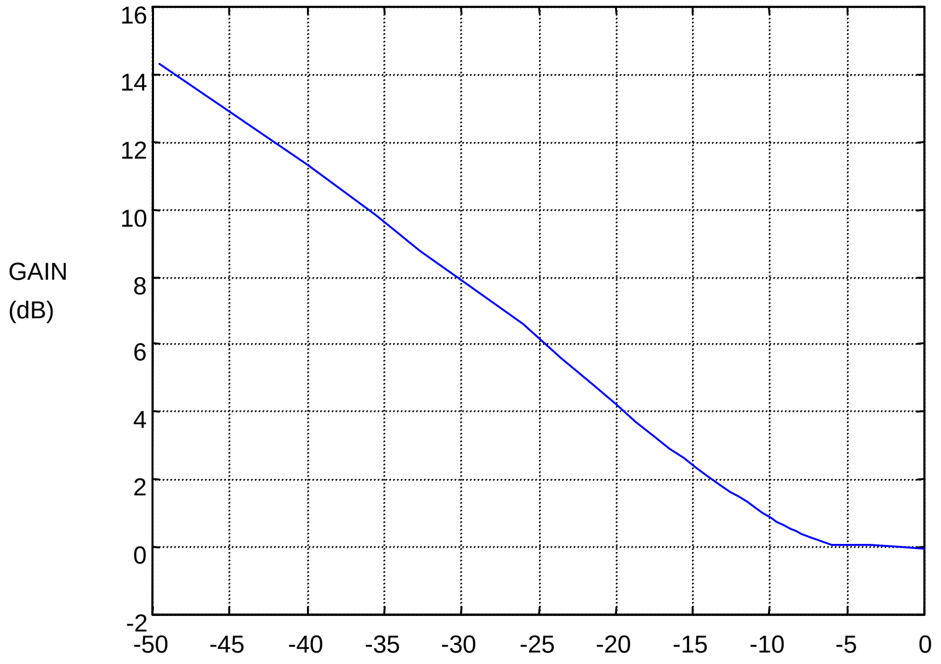
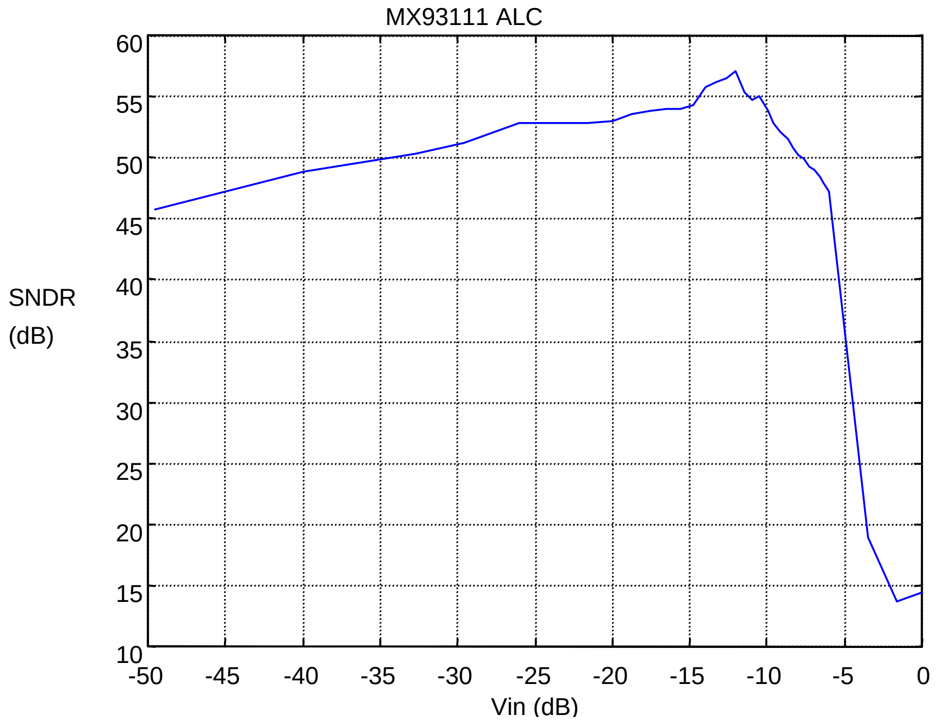

GAIN
(dB)
16
14
12
10
8
6
4
2
0
-2
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0

FIG. 7


MX93111 ALC
SNDR
(dB)
60
55
50
45
40
35
30
25
20
15
10
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0
Vin (dB)

8.0 ORDERING INFORMATION

PART NO	PACKAGE TYPE
MX93132	PQFP

MX 93 132 F C

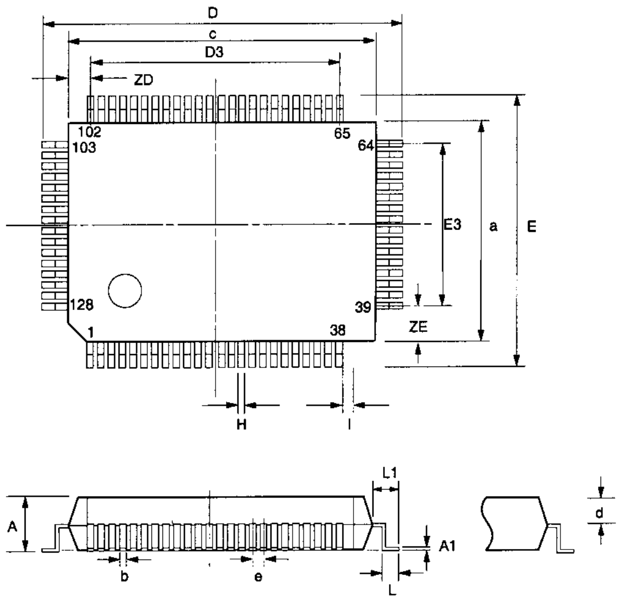
- C — COMMERCIAL 0 ~ 70 ^{O}C
- F — PACKAGE TYPE F : PQFP
- 132 — PRODUCT NUMBER
- 93 — FAMILY PREFIX
- MX — MXIC COMPANY PREFIX

8.1 PACKAGE INFORMATION for 128 PIN PQFP

128-Pin Plastic Quad Flat Pack

ITEM	MILLIMETERS	INCHES
a	14.00±.05	5.512±.002
b	.20 [Typ.]	.08 [Typ.]
c	20.00±.05	7.87±.002
d	1.346	.530
e	.50 [Typ.]	.20 [Typ.]
L1	1.60±.1	.63±.04
L	.80±.1	.31±.04
ZE	.75 [Typ.]	.30 [Typ.]
E3	12.50 [Typ.]	4.92 [Typ.]
E	17.20±.2	6.77±.08
ZD	.75 [Typ.]	.30 [Typ.]
D3	18.50 [Typ.]	7.28 [Typ.]
D	23.20±.2	9.13±.08
A1	.25±.1 min.	.01±.04 min.
A	3.40±.1 max.	1.34±.04 max.
Note	Short Lead	Short Lead

NOTE: Each lead centerline is located within .25 mm[.01 inch] of its true position [TP] at maximum material condition.



MACRONIX INTERNATIONAL CO., LTD.

HEADQUARTERS:

No. 3, Creation Road III, Science-Based Industrial Park, Hsin Chu, Taiwan, R.O.C.

TEL:+886-3-578-8888

FAX:+886-3-578-8887

TAIPEI OFFICE:

12F, No. 4, Min-Chuan E.Rd., Sec 3, Taipei, Taiwan, R.O.C.

TEL:+886-2-2509-3300

FAX:+886-2-2509-2200

EUROPE OFFICE:

Grote Winkellaan 95, Bus 1 1853 Strombeek, Belgium

TEL:+32-2-267-7050

FAX:+32-2-267-9700

SINGAPORE OFFICE:

5 Jalan Masjid Kembangan Court #01-12 Singapore 418924

TEL:+65-747-2309

FAX:+65-748-4090

MACRONIX AMERICA, INC.

1338 Ridder Park Drive, San Jose, CA95131 U.S.A.

TEL:+1-408-453-8088

FAX:+1-408-453-8488

JAPAN OFFICE:

NFK Kawasaki Building, 8F, 1-2 Higashida-cho, Kawasaki-ku

Kawasaki-shi, Kawasaki-ken 210, Japan

TEL:+81-44-246-9100

FAX:+81-44-246-9105