SGS-THOMSON MICROELECTRONICS

ST18933

DATA SIGNAL PROCESSOR

PRELIMINARY DATA

THE ST18933 INCLUDES :

- A ST18932-CORE (CPU)
- 4K X 16-BIT DATA RAM
- 8K X 16-BIT DATA ROM
- 16K X 32-BIT PROGRAM ROM
- A BOOTSTRAP LOGIC USED TO LOAD THE EXTERNAL PROGRAM MEMORY FROM THE HOST INTERFACE.
- A 8-BIT HOST INTERFACE WITH A 64 X 8-BIT DUAL PORT RAM
- 3 SYNCHRONOUS SERIAL INTERFACE
  - TWO DOUBLE-BUFFERED SSI
  - ONE SIMPLIFIED SYNCHRONOUS SERIAL INTERFACE (SSSI)
- A 4-BIT GENERAL PURPOSE PARALLEL PORT
- TWO MEMORY PAGING REGISTERS
- AN EXTERNAL DATA BUS (48K X 16-BIT)

DESCRIPTION

The ST18933 is a standard DSP based on the SGS-THOMSON ST18932-CORE (see the ST18932 CORE & EMULATION CHIP SPECIFICATION). This DSP has been designed specifically for communications applications. It excells, by virtue of its complex multiplier mode, in such fields as digital filtering, echo cancellers, equalisers and FFT operations.

The large amount of onchip program memory (16K x 32) means complete systems can be integrated into 2 chips (DSP + Analog front end). External memory (up to 64K x 32) can be used in conjunction with the internal memory to integrate large and very complex systems - or can give an easy path to upgrading evolving systems without hardware changes.

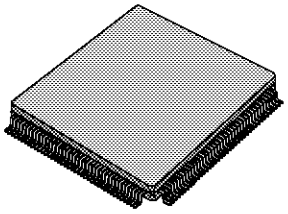
Dedicated communications system peripherals give the optimum performance/features balance for such applications as V.32 bis (or equivalent modems) ... etc.

This product is supported by a PC development tool, in the form of a slot in PC card, for IBM PCS.

Software includes Macro assembler, linker and real time emulation.

Hardware gives a code development system enabling one to down load and execute offline files from a PC.

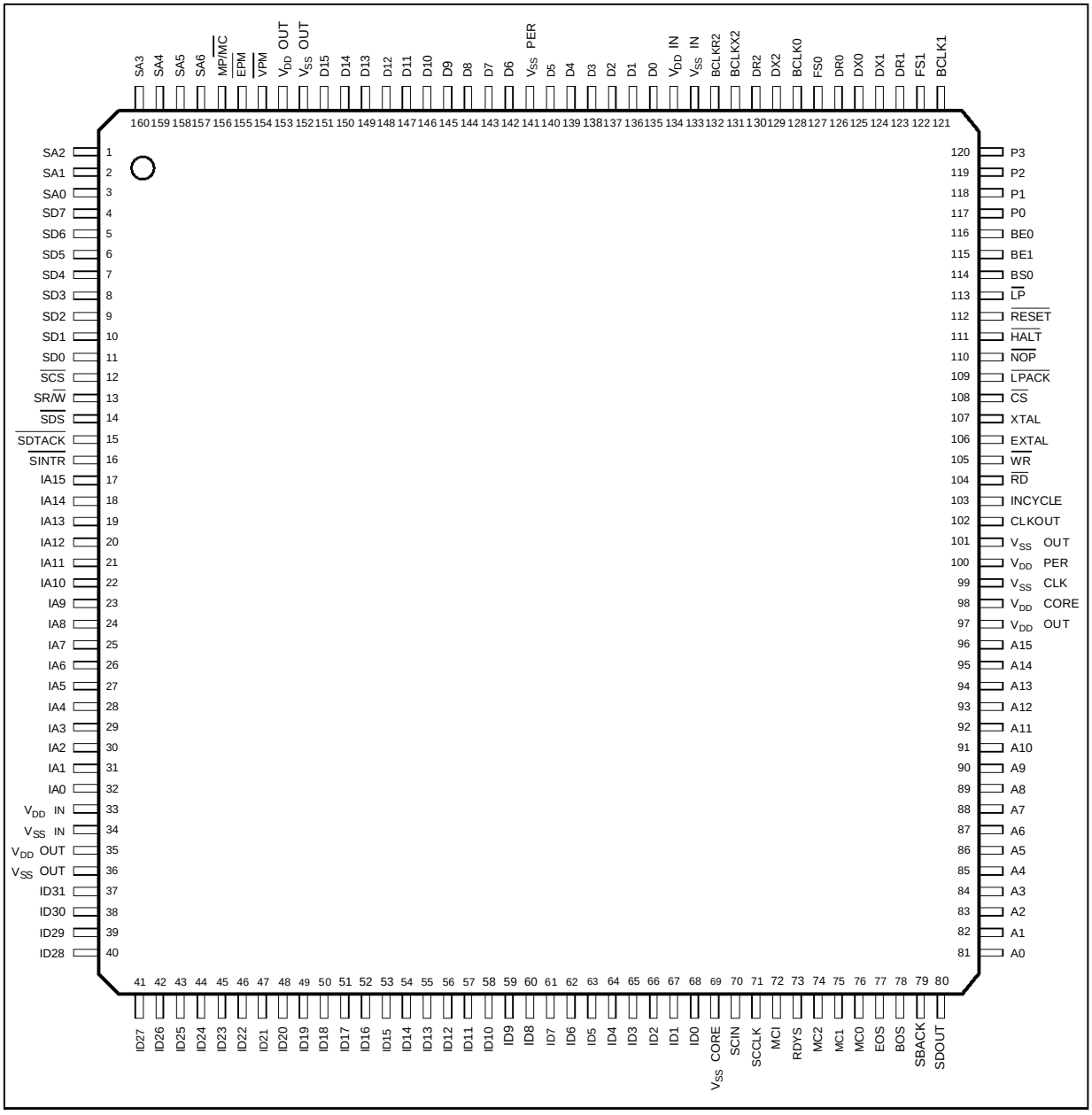
As the ST18933 is based on the ST18932 DSP card for further reading the ST18923 Technical manual is recommended (reference code ST18932TM/2).



PQFP160
(Plastic Package)

ORDER CODE : ST18933

PIN CONNECTIONS



PIN DESCRIPTION

Name	Pin Type	Function	Description	Nbr
LOCAL DATA BUS				
D0-D15	I/O	ST18932-CORE Data bus	16-bit data bus. High Impedance when exchanges are not active or when RESET or LP are active.	16
A0-A15	O	16-bit address bus for the local data bus	12-bit ST18932-core address bus and 4-bit page register. High level ($FFFF) when exchanges are not active.	16
$\overline{CS}$	O	External Space selection	Active low when address > 16k	1
$\overline{RD}$ ($\overline{DS}$)	O	Read or Data Strobe active low	RD synchronizes the read cycles. DS synchronizes all the exchanges. High level when inactive.	1
$\overline{WR}$ (R/$\overline{W}$)	O	Write or Read/Write	WR synchronizes the write cycles (active low). R/W indicates the direction of the exchange. High level when inactive.	1
INSTRUCTION BUS				
ID0-ID31	I/O	Instruction Data bus	32-bit data bus used to read instructions from the program memory or write instructions from the Host interface into the program memory.	32
IA0-IA15	O	Instruction address bus	16-bit address bus for external program memory.	16
MP/$\overline{M}$C	I	Microprocessor / Microcontroller Mode Select	High level selects MP mode : use only external program memory. Low level selects MC mode : use internal and external program memory.	1
$\overline{EPM}$	O	External Program Memory Enable	When low, enables External Program Memory use	1
$\overline{WPM}$	O	Write External Program Memory, active low	Enables a write in the external program memory	1
BRANCH / INTERRUPTS				
BS0, BE0, BE1	I	Branch conditions / Interrupt sources	Individually testable Branch conditions or maskable Interrupt sources. BS0 is low level sensitive. BE0 and BE1 are falling edge sensitive	3
HOST INTERFACE				
SD0-SD7	I/O	System Data bus	8-bit data bus used for asynchronous exchanges between the ST18933 and a host. High Impedance when exchanges are not active.	8
SA0-SA6	I	System address bus	7-bit address bus for the system data bus.	7
$\overline{SDS}$	I	Data Strobe active low	SDS synchronizes all the exchanges	1
SR/$\overline{W}$	I	Read/Write	Active low, asynchronous	1
$\overline{SCS}$	I	Chip Select active low		1
$\overline{SDTACK}$	O	System bus Data ACKnowledge		1
$\overline{SINTR}$	O	Interrupt Request, active low	Asserted by the DSP and negated by the host.	1
TWO DOUBLE-BUFFERED SYNCHRONOUS SIO; (SSI)				
FS0/FS1	I/O	Transmit and receive Frame Synchro for SSI0 and SSI1		2
BCLK0 BCLK1	I/O	Transmit and receive Clock for SSI0 and SSI1		2
DX0/DX1	O	Transmit data for SSI0 and SSI1		2
DR0/DR1	I	Receive data for SSI0 and SSI1		2
SIMPLIFIED SYNCHRONOUS SERIAL INTERFACE (SSSI)				
BCLKX2	I	Transmit Clock		1
BCLKR2	I	Receive Clock		1
DX2	O	Transmit data		1
DR2	I	Receive Clock		1

18933-01.TBL

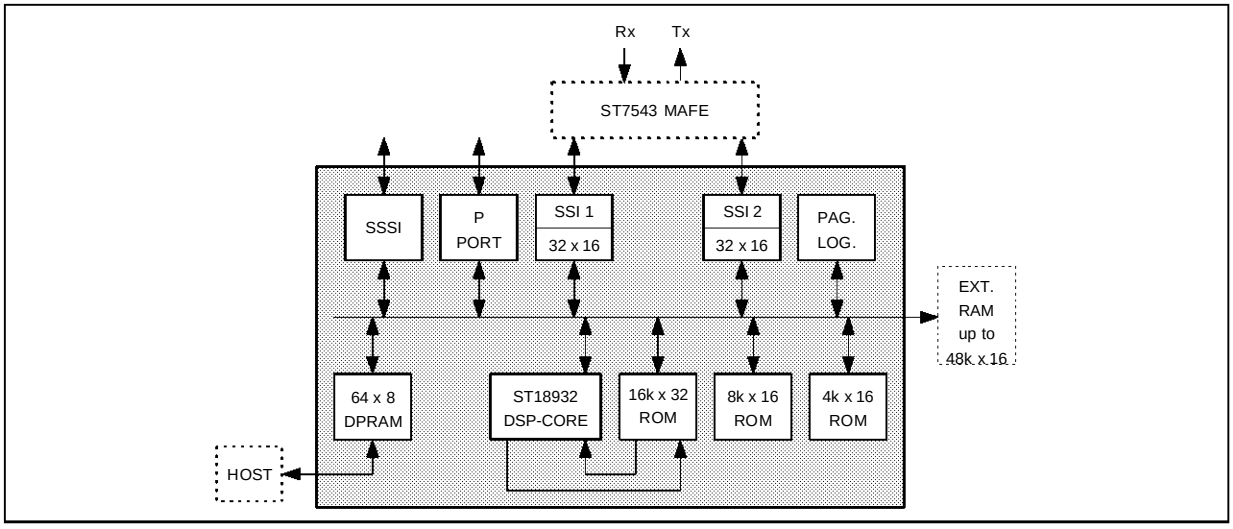
PIN DESCRIPTION (continued)

Name	Pin Type	Function	Description	Nbr
GENERAL PURPOSE 4-BIT PARALLEL PORT				
P0-P3		4-bit General purpose Parallel port	Each bit is individually programmable as an input or an output.	4
MISCELLANEOUS				
$\overline{\text{RESET}}$	I	Reset, active low	Forces the DSP into its initial state ; the program counter is loaded with the value 0 ; the instruction NOP is executed ; the clock generator is resynchronized.	1
$\overline{\text{LP}}$	I	Low Power, active low	Can be programmed as low level or falling edge sensitive ; stops the DSP at the end of the next instruction ; the NOP instruction is executed ; puts the DSP in low power state: internal DSP clock halted. The internal processor state is kept.	1
$\overline{\text{LPACK}}$	O	Low Power ACKnowledge	Is asserted low when the DSP enters into low power mode ; remains low until the DSP exits this mode.	1
$\overline{\text{HALT}}$	I	halt, active low.	Forces a Halt at the end of the next instruction. The IA0-IA15 bus is asserted in high impedance	1
$\overline{\text{NOP}}$	I	No Operation, active low	A NOP instruction is executed at the end of the next instruction.The buses IA0-IA15 , ID0-ID31 , A0-A15 , D0-D15 are asserted in high impedance.	1
CLOCKS AND POWER SUPPLY				
EXTAL	I	DSP Clock input	External clock or internal oscillator input. EXTAL frequency is twice the machine cycle frequency.	1
XTAL	O	Crystal output	Internal oscillator output for crystal. Not connected if the internal oscillator is not used.	1
CLKOUT	O	Internal DSP Clock output (machine cycle)	The frequency equals EXTAL divided by 2.	1
INCYCLE	O	Instruction Clock output	A new instruction execution starts on INCYCLE rising edge.	1
V_{DD}		Power supply	5 Volts	7
V_{SS}		Ground	0 Volt	8
BOUNDARY SCAN AND EMULATION INTERFACE - These pins are dedicated to the test and the emulation (see the ST18932 CORE & EMULATION CHIP SPECIFICATION).				
SCCLK	I	Scan CLock	Boundary-scan clock.	1
SCIN	I	Scan data Input	Boundary-scan register data input.	1
SCOUT	O	Scan data Output	Boundary-scan register data output.	1
BOS	I	Begin of Scan control	Enables data capture on the DSP-CORE I/O pins.	1
EOS	I	End Of Scan control	Enables update of DSP-CORE input data with Boundary-scan register.	1
MC0-MC2	I	Boundary-Scan Mode Control	Define the Boundary-Scan Test and Emulation Modes. The mode change occurs on the SCCLK rising edge.	3
RDYS	O	Ready to Scan flag	Set to initialise the data capture by the emulator when in Boundary-Scan Emulation Modes.	1
SBACK	O	Software Breakpoint Acknowledge	Set while executing a Software Breakpoint Instruction for Emulator use.	1
MCI	O	MultiCycle Instruction.	When low, indicates to the emulator, the execution of the first cycle of a multicycle instruction (BRANCH instruction or instruction in CPLX or DBPR mode). High level during the execution of the second cycle of a multicycle instruction or during the execution of a single cycle instruction.	1

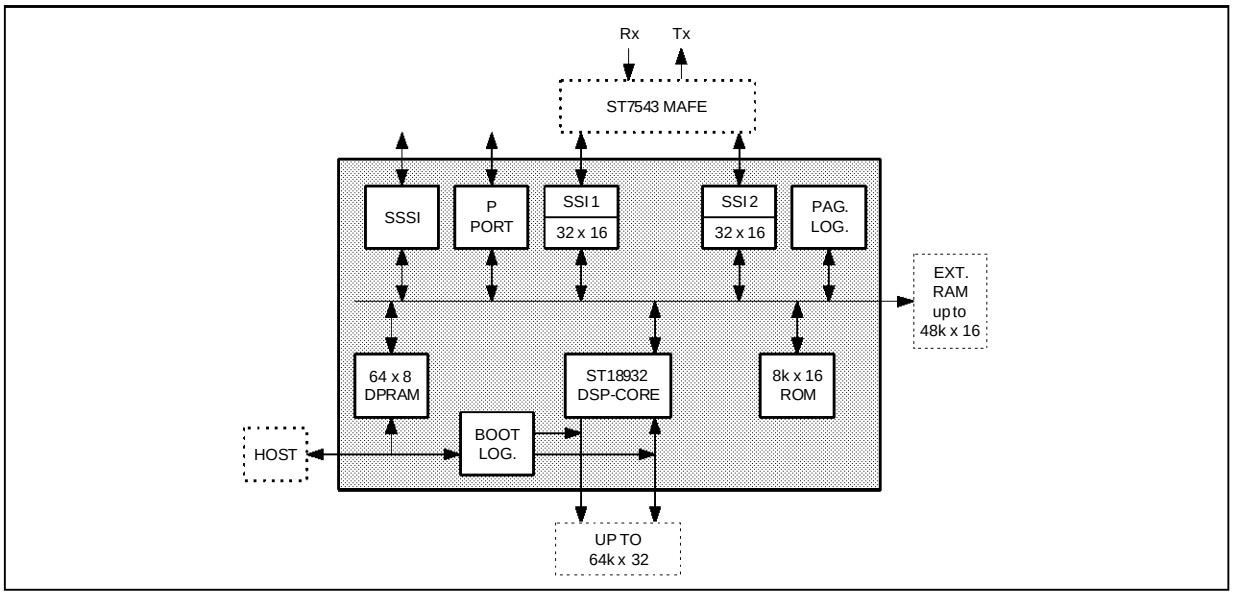
18933-02.TBL

Note : 5 DSP-CORE BRANCH/INTERRUPT INPUTS are internally used by the communication peripherals.
- BS1 is used by the MICRO interface (level sensitive)
- BS2 SSSI (transmit) (level sensitive)
- BS3 SSSI (receive) (level sensitive)
- BE2 SSI0 (transmit and receive) (edge sensitive)
- BE3 SSI1 (transmit and receive) (edge sensitive)

ST18933 BLOCK DIAGRAM



ST18R933 BLOCK DIAGRAM



ST18933 HARDWARE DESCRIPTION

1 - ST18932-CORE

(See the ST18932-CORE & EMULATION CHIP SPECIFICATION)

The ST18932-CORE is the SGS-THOMSON Digital Signal Processor macrocell.
The ST18932-CORE key features are :
- 60 ns machine cycle
- ST18 family architecture
- 16 x 16 ⇒ 32, signed and unsigned multiplier
- 32-bit barrel shifter
- 32-bit Arithmetic and Logic Unit
- provision for floating point operations
- 4 x 32-bit accumulators
- 4-level 32-bit FIFO
- Immediate and computed branch
- 2-level stack
- 8 external interrupt sources
- Automatic loops, up to 256 times 32 instructions
- 3 independent address calculation units
- Addressing modes: immediate,direct,indirect with post modification,circular
- 192 x 16-bit and 128 x 16-bit on-chip Data RAM
- 8k of linear addressing space (16-bit local data bus)
- Software wait-states to access slower external memory/peripheral
- access to 64k x 32-bit program memory
- Powerdown mode
- Boundary-Scan features for test and emulation
- VHDL IEEE Standard 1076 model for simulation

2 - MEMORY AND ADDRESSING SPACE

All the on-chip communication peripherals and the internal data memory (except the X and Y RAM which are into the DSP-CORE) are mapped in the E and C space of the DSP-CORE.

The access is controlled by the signals:
- $\overline{RD}$ which synchronizes the read cycles
- WR which synchronizes the write cycles

> In order to access to the internal space (16k : address $0000 to $ 3FFF) the DSP Access Mode Register (AMR bit 3 (IM) MUST BE SET TO 0.

The external space can be controlled by $\overline{RD}$ and WR or DS and R/W signals.

> The internal data memory and the communication peripherals are accessed at full speed, except the Dual Port Ram which requires 2-cycle exchanges

The ST18933 provides:
- 4k x 16-bit internal high speed data RAM
- 8k x 16-bit internal high speed data ROM
- 16k x 32-bit internal high speed program ROM
- access to 48k x 16-bit external data memory
- access to 64k x 32-bit external program memory

The amount of data memory leads to use two page registers :
- 1 x 4-bit register for E-space
- 1 x 4-bit register for C-space

Then, each page is 4k x 16-bit large (A0-A11 address bits), and the page registers contain the MSB (A15-A12) of the address bus. These page registers are mapped in the C-space, at the address :
- xFFE ⇒ E-paging register
- xFFF ⇒ C-paging register

"x" means that the registers are duplicated in each page of the C-space.

The page registers can be written and read by the DSP-CORE.
They are LSB-adjusted regarding the local bus (D3-D0), and the bits D15-D4 are asserted low during a read cycle.

Except the two paging register addresses in the C-space, the 5 pointer registers (E0,E1,C0,C1,C2) address the same memory locations (E/C signal is not used in the address decoding).

The C-page register is selected when E/C=0 i.e. when C0 or C1 or C2 pointer is selected.
The E-page register is selected when E/C=1 i.e. when E0 or E1 or no pointer is selected.

This architecture allows work with two sets of data located in two different pages without time penalty. Furthermore, the on chip memory sharing between the E-space and the C-space depends only on the software in order to fit the application.

The communication peripherals are mapped in the page 0.
The data memory mapping is defined in the following table (see also page 0 mapping in section 6):

	E space	C space
Page 0	Communic. Periph.	Communication Peripheral + Paging
Page 1	Int. RAM 4k x 16-bit	Int. RAM 4k x 16-bit + Paging
Page 2	Int. ROM 8k x 16-bit	Int ROM 8k x 16-bit + Paging
Page 3		
Page 4 to Page 15	Ext. RAM	Ext. RAM + Paging

While the DSP transfers data to/from the pages 0,1,2 or 3 :
- the data bus D0-D15 is in high impedance state;
- the address bus A0-A15 is asserted to FFFF;
- the Chip Select signal $\overline{CS}$ is inactive (high level).

3 - COMMUNICATION PERIPHERALS

3.1 - Host Interface : Dual Port RAM

3.1.1 - Introduction

The ST18933 exchanges information with the host processor through a general purpose bus. A 64 x 8-bit Dual Port Ram is used as a buffer between the host and the DSP-CORE.

Regarding the host :

The ST18933 is seen as 128 x 8-bit memory locations.

The Dual Port Ram is located from address $00 to address $3F.

The registers are located from address $40 to address $54.

The registers are LSB-adjusted regarding the system bus (SD0-SD7).

Then, all the unused MSB are read at 0 by the host processor.

Regarding the DSP-CORE :

The host interface is mapped in the page number 0 of the E/C-space (See sections 2 and 6).

The Dual Port Ram is located from address $0000 to address $003F.

The registers are located from address $0040 to address $0054.

The data coming from the Dual Port Ram and the registers are LSB-adjusted regarding the local bus (D0-D15).

Then, all the unused MSB are read at 0 by the DSP-CORE.

3.1.2 - Pin Description

20 pins are used to interface with the host processor.

SD0-SD7 : 8-bit wide bidirectionnal data bus

SA0-SA6 : 7-bit wide input address bus

SDS : input data strobe

SR/W : this input pin indicates the direction of the exchange.

SCS : input. Chip Select

SDTACK : DATA ACKNOWLEDGE. This pin is asynchronous and active low

SINTR Interrupt Request active low. This output(which may be wired to the interrupt pin of the host processor) is activated by the DSP and desactivated by the host processor.

3.1.3 - Dual Port RAM

The dual port RAM is a 64 x 8-bit static memory.

This memory is shared by the host processor and by the DSP and is mapped :

- from address 00 to 3F in the host space
- from address 0000 to 003F in the ST18933 space.

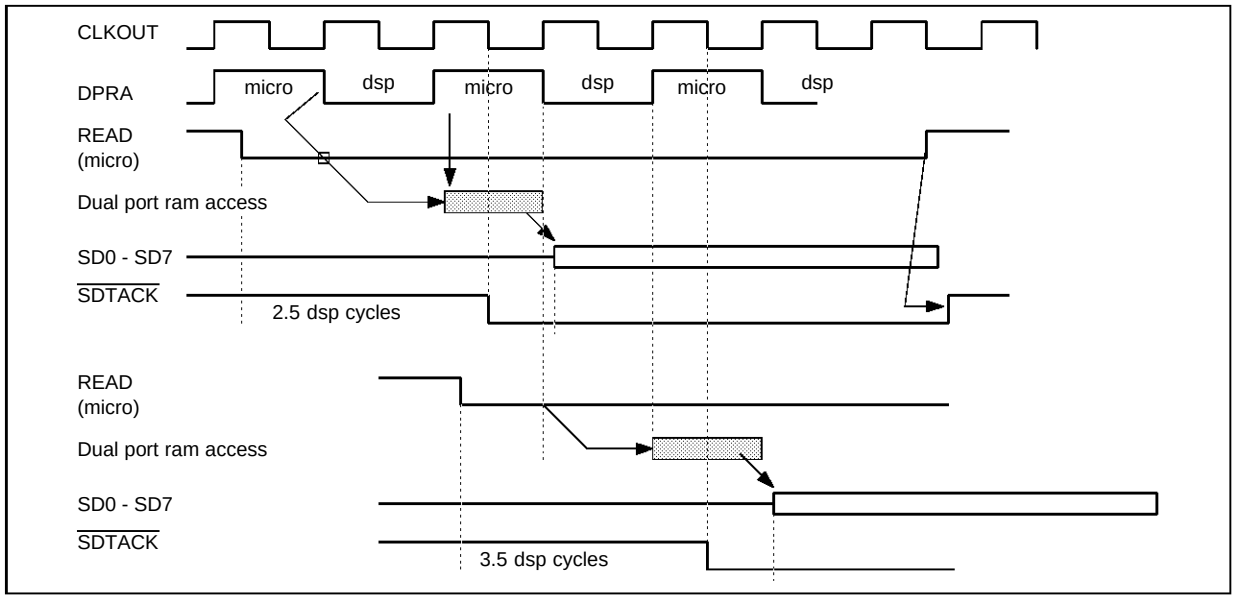
The access to the memory is synchronous with the DSP instruction cycle .It's time shared between the host microprocessor and the DSP.

CLKOUT, output clock of the DSP, is divided by two to give the signal DPRA (Dual Port Ram Access).

When DPRA is high, the Dual Port Ram belongs to the host processor, when DPRA is low, the Dual Port Ram belongs to the DSP.

Under these conditions, the worst case latency is TWO DSP cycles time for DSP access, and FOUR DSP cycle time for the host processor access (see Figure 1).

Figure 1



3.1.4 - Interrupt Control (see Figure 2)

In association with the Dual Port Ram, a bidirectionnal interrupt controller is implemented:

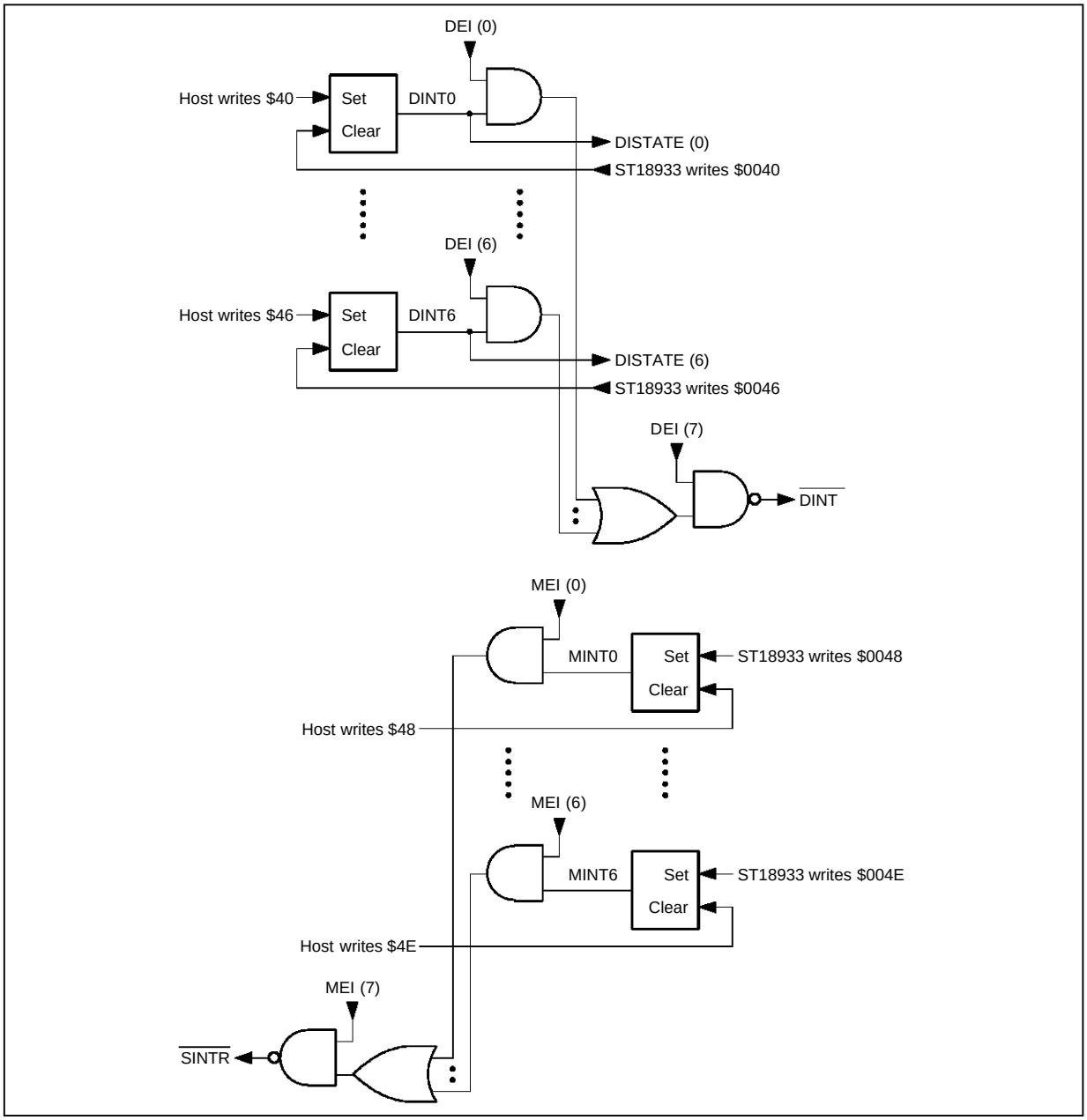
- the ST18933 may generate 7 interrupt sources for the host processor,
- the host processor may generate 7 interrupt sources for the ST18933.

The interrupt controller characteristics are:

- dual sided interrupt interface
- single interrupt signal (SINTR for the micro, DINT for the DSP-CORE), inclusive OR between 7 interrupt sources
- every interrupt source generated by the DSP is maskable thanks to MEI register
- every interrupt source generated by the micro is maskable thanks to DEI register
- SINTR is set by the DSP and reset by the host processor
- DINT is set by the micro and reset by the DSP

Notes : 1. In case of simultaneous set and reset of an interrupt, the interrupt is inactive.
2. DINT drives the DSP-CORE input BS1: branch/interrupt on level (low level sensitive).

Figure 2



The interrupt sources, the mask registers and the status registers are mapped from address $40 to address $52, as mentioned in the figure below :

Micro space		ST18933 space
$00 : : $3F	**64 x 8-bit DUAL PORT RAM**	**$0000 : : $003F**
$40	MINT0	$0040
$41	MINT1	$0041
$42	MINT2	$0042
$43	MINT3	$0043
$44	MINT4	$0044
$45	MINT5	$0045
$46	MINT6	$0046
$47	Unused	$0047
$48	DINT0	$0048
$49	DINT1	$0049
$4A	DINT2	$004A
$4B	DINT3	$004B
$4C	DINT4	$004C
$4D	DINT5	$004D
$4E	DINT6	$004E

Micro space		ST18933 space
$00 : : $3F	**64 x 8-bit DUAL PORT RAM**	**$0000 : : $003F**
$4F	MEI	$004F
$50	MISTATE	$0050
$51	DEI	$0051
$52	DISTATE	$0052

MINT0 to MINT6 are the 7 interrupt sources generated for the host processor, by the ST18933.

MINT0 - MINT6 are :
- set when the ST18933 writes at the corresponding addresses ($0040 to $0046)
- cleared when the host processor writes at the same addresses.

DINT0 to DINT6 are the 7 interrupt sources generated for the DSP by the host processor.

DINT0 - DINT6 are :
- set when the host writes at the corresponding addresses ($48 to $4E)
- cleared when the ST18933 writes at the same addresses.

MEI: 8-bit mask register

7	6	5	4	3	2	1	0
Global Enable	Enable MINT6	Enable MINT5	Enable MINT4	Enable MINT3	Enable MINT2	Enable MINT1	Enable MINT0

This register is : read and written by the host processor (address $4F)
read only by the ST18933 (address $004F)

MEI(7): is the global interrupt mask.
MEI (7) = 0 → interrupts are not allowed
MEI (7) = 1 → interrupts are allowed

MEI(i) (with i= 0,1,2,3,4,5,6) is the mask of MINTi.
MEI (i) = 0 → MINTi is disabled
MEI (i) = 1 → MINTi is enabled

MISTATE: 7-bit register indicating the interrupt source

6	5	4	3	2	1	0
State of MINT6	State of MINT5	State of MINT4	State of MINT3	State of MINT2	State of MINT1	State of MINT0

This register is : read by the host processor (address $50)
read by the ST18933 (address $0050)

MISTATE(i) = 0 → MINTi is active
MISTATE(i) = 1 → MINTi is inactive

The bits of MISTATE Register do not take their corresponding mask bit (MEI[i]) into account.

DEI: 8-bit mask register

7	6	5	4	3	2	1	0
Global	Enable	Enable	Enable	Enable	Enable	Enable	Enable
Enable	DINT6	DINT5	DINT4	DINT3	DINT2	DINT1	DINT0

This register is : read and written by the ST18933 (address $0051)
read only by the host processor (address $51)

DEI(7): is the global interrupt mask.
DEI(7) = 0 → interrupts are not allowed
DEI(7)= 1 → interrupts are allowed

DEI(i) (with i= 0,1,2,3,4,5,6) is the mask of DINTi.
DEI(i) = 0 → DINTi is disabled
DEI(i) = 1 → DINTi is enabled

DISTATE: 7-bit register indicating the interrupt source

6	5	4	3	2	1	0
State of DINT6	State of DINT5	State of DINT4	State of DINT3	State of DINT2	State of DINT1	State of DINT0

This register is : read by the ST18933 (address $0052)
read by the host processor (address $52)

DEI(i) = 0 → DINTi is disabled
DEI(i) = 1 → DINTi is enabled

The bits of DISTATE register do not take their corresponding mask bit (DEI[i]) into account.

When $\overline{\text{RESET}}$ occurs, all these registers (MEI,MISTATE, DEI, DISTATE) are cleared to 0. Therefore, the interrupts are masked (global mask and individual mask) and cleared.

3.1.5 - $\overline{\text{SDTACK}}$ GENERATION

$\overline{\text{SDTACK}}$ is generated at every exchange. It is activated when the Dual port RAM (or registers) access is completed.

3.2- Two Double-buffered Synchronous SIO

The two SSIs are identical to each other. The description below holds for each of them. The only difference is their addresses in the ST18933 memory map, which is covered below an also in sections 3.2 and 3.6.

	ST18933 address space
SSIO buffers : transmit if DSP writes receive if DSP reads	**$0100 : : $010F**
SSIOR1	$0110
SSIOR2	$0111
SSIOR3	$0112
Unused	: : $01FF

	ST18933 address space
SSI1 buffers : transmit if DSP writes receive if DSP reads	**$0200 : : $020F**
SSI1R1	$0210
SSI1R2	$0211
SSI1R3	$0212
Unused	: : $03FF

The SSI is synchronous only . Transmission and reception are synchronised by the same clock and frame synchronisation signals.
The SSI can be master of the clock system or slave of another device.

3.2.1- PINS

The SSI is a 4-wire interface : data in and data out are shifted by the same clock , and frame start is indicated by a single signal.

DX : data transmit
DR : data receive
BCLK : bit clock
FS : frame synchronisation

DX0, DR0, BCLK0, FS0 for SSI0
DX1, DR1, BCLK1, FS1 for SSI1

3.2.2- FORMAT COMPATIBILITY

Generally speaking, the SSI support protocols where a frame is initiated by a pulse or validated by a level on FS. In particular, the SSI will be able to exchange data with following devices using following protocols :

ST 7543

A frame consists of 4 x 16, initiated by a high pulse on FS (”short synch” or ”delayed mode”). BCLK and FS are generated by the 7543. Data in is sampled on falling edge of BCLK, and data out is issued on rising edge.

ETC 5057 and TS 5070 combos

8-bit data , short synch frame. BCLK and FS have to be supplied to the combos. Data in is sampled on falling edge of BCLK, and data out is issued on rising edge.

MOTOROLA DSP56ADC16

A frame consists of 16 bits, initiated by a high pulse on FS, or validated by a low level on FS (user programmable). Data is sampled on falling or rising edge of BCLK, respectively. BCLK and FS are generated by the DSP56ADC16.

PHILIPS I2S components

A frame consists of one word for right and one word for left channel, accompanied by a high and low level on FS respectively. Words of 16 bits are supported directly. Data in is sampled on rising edge of BCLK, and data out is issued on falling edge.

CRYSTAL A/Ds

A frame consists of 16 bits validated by a low level on FS, or a high pulse on FS, depending on the circuits. They generate BCLK and FS. Data out is issued on rising edge of BCLK.

ANALOG DEVICES D/As

A frame consists of 16, 18 or 20 bits marked by a FS high to low transition after the last bit. More bits than actually used (ie 32 instead of 18 or 20) can be shifted in without problem.

They must be given BCLK and FS. Data is sampled on rising edge of BCLK.

ST18940

A frame consists of 8, 16, 32 or 64 bits. Frame synch is a high pulse on FS (”delayed mode”).

MOTOROLA SPI

A frame consists of one 8-bit word, validated by a low level on NSS. Clock is present only during the frame. BCLK and FS can be supplied by either device, depending on which one is set to master mode. In CPHA = 1, CPOL = 0 mode , data in is sampled on falling edge of BCLK, and data out is issued on rising edge.

3.2.3 - GENERAL OPERATION

The clock phase is user programmable : data can be either sampled on rising or falling edge of BCLK. The frame synchronisation (FS) is user programmable among following modes :

- it can be a pulse, only the leading edge of which being relevant.
- it can be a level, lasting for the whole frame of data . In each case, the active edge (resp. active level) is programmable.

From the SSI point of view, the current frame will end when the programmed number of bits has been transferred or when the next leading edge on FS is detected, in the first case, and when FS has returned to its inactive level, in the second case.

During the time (if any) between the current frame is finished and the next one starts, the SSI will not receive any further data and will go high impedance. This allows the SSI to be used in a multi-driver configuration (serial highway) even using pulse mode, providing it is given the first time slots in the frame.

The number of bits per frame transmitted and received by the SSI is programmable among the values 8, 16, 32 and 64.

High pulse frame synchronisation :

The Figure 3 shows a frame adjacent to the previous one, but not adjacent to the following one (the output goes high impedance, the input is irrelevant).

Low level frame synchronisation (8-bit frame) :

The Figure 4 shows a frame as close as possible to the previous one, but not adjacent to the following one (the output goes high impedance).

Frame contents are buffered both on reception and transmission .

The SSI can be programmed so that buffers contain 1, 2 , 3 or 4 frames.

There are *two receive and two transmit buffers*, so that the ST18933 can read incoming data and write new data in one set of buffers while reception and transmission occur on the other set of buffers.

The ST18933 must assert its buffer set ”ready” before the SSI runs out of its own one and attempts to switch sets, otherwise an overrun condition is said to have occurred, which is flagged by a status bit. On overun, new incoming data is lost while previous data in transmit buffer is sent again.

Figure 3

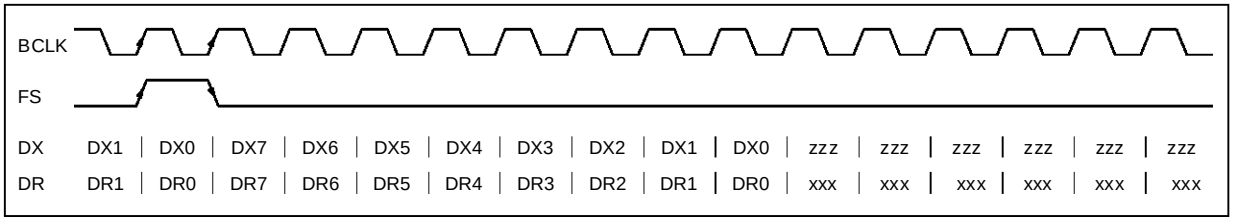
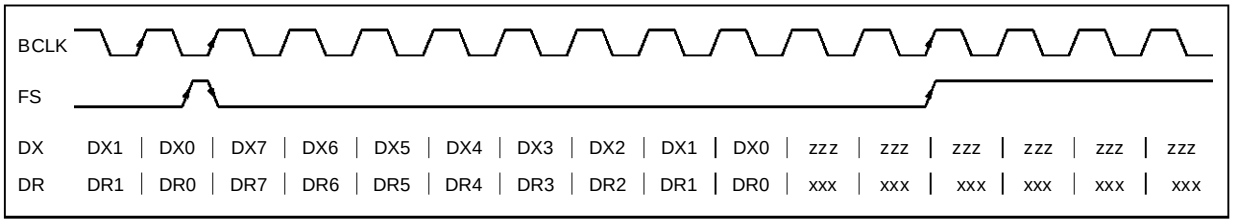


Figure 4



Buffers are organised as 16-bit wide words . In the SSI, frame 0 always starts on word 0 and frames 1, 2 and 3 (if allowed) always start on words 4 , 8 and 12 .

If the frame format is 8-bit , only the least significant byte of the first word is used .

If the format is 16, 32 or 64-bit , then 1 , 2 or 4 words starting as said upper are used .
Remaining parts of the transmit buffers are never transmitted , and the content of the same parts of the receive buffers cannot be predicted. Data is transmitted and received MSB first .

The transmit buffers are not altered by the SSI so that some words may be written once and transmitted indefinitely. In order to allow the ST18933 to update the same field of different frames by simply incrementing the address , the RAM addressing (by the DSP) may be manipulated so that words appear to the DSP in frame order within field order, whereas they are transmitted and received in field order within frame order.
It also has the benefit to squeeze all data on subsequent addresses when using 8 or 16-bit formats.
This feature is programmable by a control bit.

3.2.4 - REGISTERS
The SSI status/control Registers can be written and read by the ST18933.
They are LSB-adjusted regarding the local bus, and the unused bits are read at zero .

SSI Register 1 :

9	8	7	6	5	4	3	2	1	0
0E	IE	TA	FPB1	FPB0	BPF1	BPF0	CPHA	FST1	FST0

OE (Overrun int. Enable) : 0 → disabled
1 → enabled

IE (Interrupt Enable) : 0 → disabled
1 → enabled

TA (Twisted Addressing) : 0 → non twisted
1 → twisted

FPB1-FPB0 (Frames Per Buffer) : 00 → 1
01 → 2
10 → 3
11 → 4

BPF1-BPF0 (Bits Per Frame) : 00 → 8-bit
01 → 16-bit
10 → 32-bit
11 → 64-bit

CPHA (Clock Phase) : 0 → data starts on low BCLK
1 → data starts on high BCLK

FST1-FST0 (Frame Synchro Type) : 00 → low level
01 → high level
10 → falling edge
11 → rising edge

When $\overline{\text{RESET}}$ occurs, register 1 is cleared to 0.

SSI Register 2 :

2	1	0
OR	BSR	SE

OR (OverRun) : 0 → no overrun
1 → overrun : data is lost read-only bit. special mechanism for reset.

BSR (Buffer Set Ready) : 0 → in use by the DSP
1 → ready for use by SSI can only be written to one

SE (SSI Enable) : 0 → disabled
1 → enabled

When $\overline{\text{RESET}}$ occurs, register 2 is cleared to 0.

3.2.5 - STATUS AND CONTROL BITS USAGE

Normal operation (see Figure 5)
The Buffer Set Ready bit can only be written to one, it is cleared by the SSI when buffer sets are switched, which may generate an interrupt. Attempting to write BSR to zero doesn't change its value. From the time BSR has been written to one to the time the SSI resets it, both buffer sets are inaccessible from the DSP, and it should not attempt to do so.

Overrun (see Figure 6)
When the BSR bit has not been set at the time the SSI should switch buffer sets, the OverRun bit is set and may generate an interrupt. Once it has been read at one , the OR bit will be reset next time it is written to one, unless a new overrun condition has occurred in between.

Figure 5

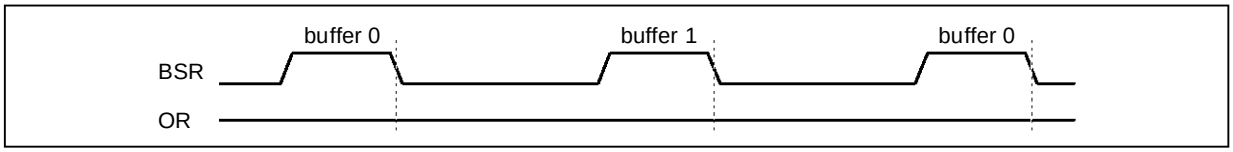
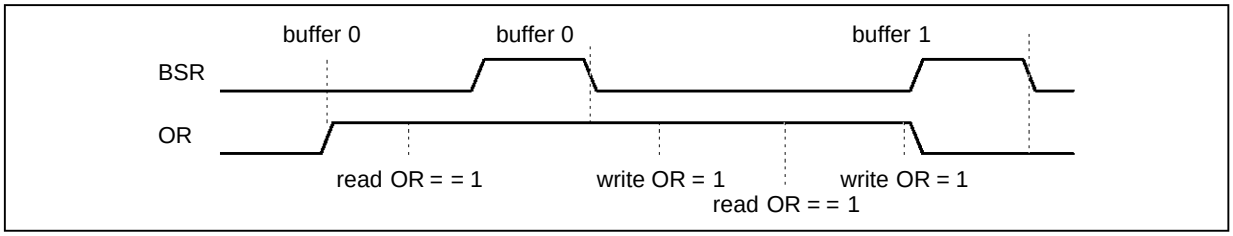


Figure 6



3.2.6- MASTERSHIP

The SSI can be given mastership of the clock and frame synch signals.
Additional control bits, located in control register 3 , are :

SSI Register 3 :

6	5	4	3	2	1	0
LP	MSTR	BSR3	BTR2	BTR2	BTR0	MFST

LP (Local Loop)	: 0	→	ormal mode
	1	→	test mode (serial Input taken from DX)
MSTR (Master)	: 0	→	slave
	1	→	master
BTR3-BTR1 (Bit Rate Ratio)	: 000	→	1/4
	001	→	1/8
	010	→	1/16
	011	→	1/32
	100	→	1/64
	101	→	1/128
	110	→	1/256
	111	→	1/512
BTR0 (Bit Rate prescaler)	: 0	→	clock/3
	1	→	
MFST (Master Frame Synch Type additional bit)			
in pulse mode	: 0	→	1 data bit long synch
	1	→	16 data bit long synch
in level mode	: 0	→	continuous operation
	1	→	intermittent operation

When $\overline{\text{RESET}}$ occurs, register 3 is cleared to 0 .

The following diagrams should clarify the situation as far as MFST is concerned :

FST = = 11 (positive pulse) and MFST == 0 (1 bit long pulse) : This is the standard short frame synchronisation (see Figure 7)

FST = = 11 (positive pulse) and MFST == 1 (half frame long pulse) : With 32 bit frames , it generates I2S format (right channel first) (see Figure 8)

FST = = 00 (low level) and MFST == 0 (continuous operation) : Data is transmitted continuously at a rate defined by the bit clock rate (8 bit frames assumed in Figure 9)

FST = = 00 (low level) and MFST == 1 (intermittent operation) : A full data buffer is transmitted each time BSR is written to one by the DSP . The data rate is defined by the software (8 bit frames and 2 frame buffers assumed in Figure 10).

Note : in pulse mode , transmission is always continuous

Figure 7

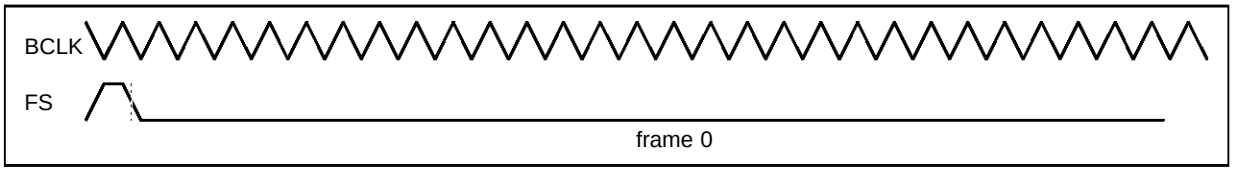


Figure 8

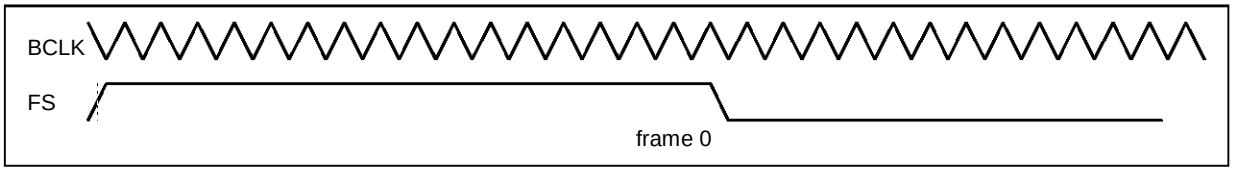


Figure 9

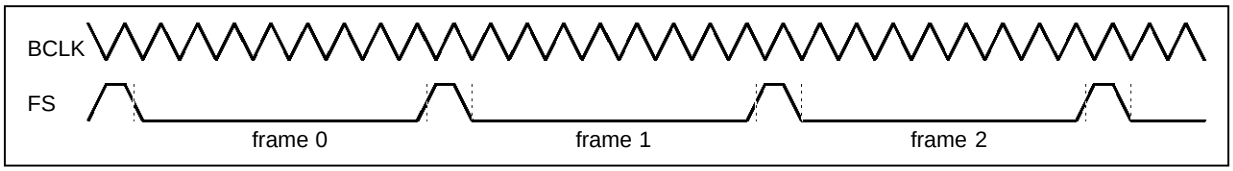
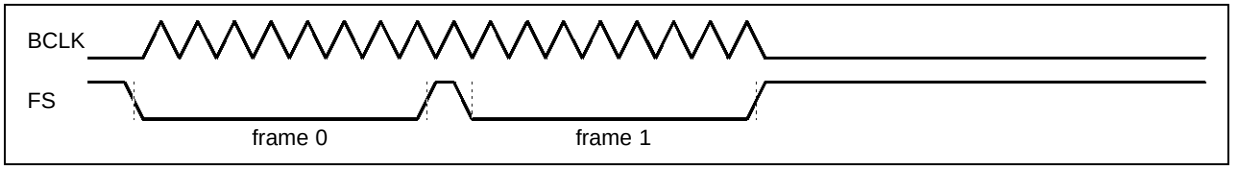


Figure 10



3.2.7- DSP-CORE INTERFACING

The SSIs are mapped in page number 0 of the DSP E/C space, from $0100 to $01FF for the first SSI, and from $0200 to $02FF for the second one.

SSI0 data buffer : $0100 to $010F
SSI0 Register 1 (SSI1R1) : $0110
SSI0 Register 2 (SSI1R2) : $0111
SSI0 Register 3 (SSI1R3) : $0112
SSI1 data buffer : $0200 to $020F
SSI1 Register 1 (SSI2R1) : $0210
SSI1 Register 2 (SSI2R2) : $0211
SSI1 Register 3 (SSI2R3) : $0212

SSI0 generates one branch/interrupt signal for the DSP-CORE input BE2 .
SSI1 generates one branch/interrupt signal for the DSP-CORE input BE3 .
BE2 and BE3 are edge sensitive.

3.3 - Simplified Synchronous Serial Interface (SSSI)

The Simplified Synchronous Serial Interface provides independent Transmit and Receive sections which use separate Clocks. Its addresses in the DSP memory map are covered in sections 3.2 and 3.6.
The SSSI is synchronous only. Transmission and reception are synchronised by two different clocks which are externally provided .
The length of the transmitted and the received messages can be different from each other (1-bit to 8-bit).

Regarding the external interface, the SSSI uses 4 pins :

- BCLKX : Transmit bit Clock (input)
- BCLKR : Receive bit Clock (input)
- DX : Transmit data (output)
- DR : Receive data (input)

The ST18933 corresponding pins are : BCLKX2, BCLKR2, DX2, DR2.
The SSSI is mapped in page 0 of the ST18933 address space.
It can generate 2 branch/interrupt signals on BS2/BS3 DSP-CORE inputs.

4 registers provide serial data transfer and parallel/serial data conversion :

- Serial input register SR (8-bit) associated with Receive Input register SIN (8-bit).
- Serial output register SX (8-bit) associated with Transmit output register SOUT (8-bit).

SR and SX registers are tied to the DR,DX pins. SIN can be read and SOUT can be loaded by the ST18932 DSP-CORE through the local bus.
The 8-bit receive register is LSB-adjusted regarding the local bus (D7-D0 used), and the bits D15-D8 are read at 0 by the ST18933.

2 Control Registers : TCR,RCR
They can be written and read by the DSP. They are LSB-adjusted regarding the local bus, and the unused bits are read at zero.

TCR : Transmit Control Register

6	5	4	3	2	1	0
$\overline{\text{DXTS}}$	$\overline{\text{DXS1}}$	TCP	TIE	TBPW2	TBPW1	TBPW0

TBPW2-TBPW0 (Transmit Bit Per Word) : 000 → 8-bit
001 → 1-bit
010 → 2-bit
011 → 3-bit
" "
111 → 7-bit

TIE (Transmit Interrupt Enable : 0 → disabled
1 → enabled

TCP (Transmit Clock Phase) : 0 → data starts on rising edge of BCLKX
1 → data starts on falling edge of BCLKX

$\overline{DXS1}$ (DX Set to 1) : 0 → DX Set to 1 (if DXTS=1)
1 → DX = SX (if DXTS=1)

$\overline{DXTS}$ (DX Tristate) : 0 → DX = SX (if DXTS=1)
1 → DX Low Impedance

When $\overline{RESET}$ occurs, register TCR is cleared to 0.

RCR : Receive Control Register

6	5	4	3	2	1	0
TDERD	CLKI	RCP	RIE	RBPW2	RBPW1	RBPW0

RBPW2-RBPW0 (Received Bit Per Word) : 000 → 8-bit
001 → 1-bit
010 → 2-bit
011 → 3-bit
" "
111 → 7-bit

RIE (Receive Interrupt Enable) : 0 → disabled
1 → enabled

RCP (Receive Clock Phase) : 0 → data sampled on rising edge of BCLKR
1 → data sampled on falling edge of BCLKR

CLKI (Internal clock) : 0 → external clocks (normal mode)
1 → test mode
- receive clock taken from SSI0
- transmit clock taken from SSI1

TDERD (Transmit data equal receive data) : 0 → normal mode
1 → test mode (serial input) taken from DX

When $\overline{RESET}$ occurs, register RCR is cleared to 0.

GENERAL OPERATION

The clock phase is user programmable : data can be either sampled on rising or falling edge of BCLKX/BCLKR.

Serial data transmission is initiated when DXTS,DXS1 are set high and when the transmit counter restarts with the value of TBPW2-TBPW0 previously loaded in the TCR register. The 8-bit data word loaded in SOUT is transfered to the SX register and clocked out of SX to DX on the rising/falling edge of BCLKX. Serial words are transmitted LSB first (LSB refers to the data loaded in SOUT from the local bus bit D0).

Upon completion of the SX to DX transfer , the DSP-CORE input BS2 is asserted low (If Interrupt Enable bit is set). The transmit branch/interrupt source BS2 is removed by loading a new value in SOUT.The data transfert from SOUT to SX register occurs when the current word has been transfered to DX. Until SX has been loaded the same word is transfered to DX.

Serial receive data is always clocked in SR register.

When the transfer from DR to SR is completed, depending on the value of the receive counter, the SIN register is loaded and an interrupt is generated (if RIE bit is set). The receive interrupt source BS3 is removed by reading the value in SIN. Every new word,

SIN is loaded even if data has not yet been read by the DSP_CORE.

Writing a new value in TCR/RCR to change the word's length takes effect when the current word transmission or reception is completed, except for TCP and RCP bit which take immediatly effect.

Provision must be taken before changing RBPW2-RBPW0/TBPW2-TBPW0 (number of bits per word) during a valid word receive/transmit.

The new word length becomes valid either for the next word or the word after the next one (see diagrams) depending on the state of the SSSI clock when the update occurs.

DSP-CORE INTERFACING

The SSSI is mapped in page 0 of the ST18933 space, from the adress $0400.

SSSI Data Transmit Register (SOUT) : $0400
SSSI Data Receive Register (SIN) : $0401
SSSI Transmit Control Register (TCR) : $0402
SSSI Receive Control Register (RCR) : $0403

When TIE and RIE are set high the SSSI generates 2 branch/interrupt signals for the DSP-CORE:
- transmit → BS2 input
- receive → BS3 input

3.4- General Purpose 4-bit Parallel Port (PPORT)

The general purpose 4-bit Parallel port (PPORT : P0-P3) consists of :
- regarding the DSP-CORE : a 4-bit read/write register mapped in the page number 0 of the E/C-space at the address $0800 ;

PPORT :

3	2	1	0
P3	P2	P1	P0

- regarding the peripheral: 4 independant input/output lines.

Each bit can be individually programmed as an input or an output through a 4-bit Parallel Port Direction Register (PPDR) controlled by the DSP and located in the same page number 0 (address $0801).

PPDR : PPort Data Direction :

3	2	1	0
PPDR3	PPDR2	PPDR1	PPDR0

The parallel port and the PPDR registers are read/write registers, LSB-adjusted regarding the local bus (D3-D0 used), and the bits D15-D4 are asserted to 0 during a DSP read cycle (see Figure 11).

When a bit is programmed as an output:
- the corresponding ST18933 pin is the mirror of the related PPORT output register and is always in low impedance;
- the DSP can read back the output register.

When a bit is programmed as an input:
- the corresponding ST18933 pin state is written, each CLKOUT cycle, into the related PPORT input register and the DSP can read it;
- the DSP can still write to the output register.

The DSP accesses the 4-bit PPORT globally.

When $\overline{\text{RESET}}$ occurs:
- all the PPDR bits are set to 0
- each PPORT bit is configured as an input and cleared to 0.

4 - WRITE (Bootstrap Function) / READ EXTERNAL PROGRAM MEMORY

The bootstrap function allows a program to be loaded from the host interface downto the external program memory.

In order to check the DSP external program memory contents, the ST18933 provides the host with a read function.

These read/write functions are fully controlled by the host processor ($\overline{\text{SDS}}$,SR/$\overline{\text{W}}$,$\overline{\text{SCS}}$,$\overline{\text{SDTACK}}$, signals and SD0-SD7,SA0-SA6 buses).

The host interface provides a 128 x 8-bit memory space:
- the Dual Port RAM is located at the 64 lowest addresses
- the following are the command registers of the interface
- the bootstrap device is located at the highest addresses.
- Each instruction needs 6 bytes:
- 2 bytes for the 16-bit address
- 4 bytes for the 32-bit data

The bootstrap device builds the 16-bit and 32-bit words (see Figures 12 and 13).

Figure 11

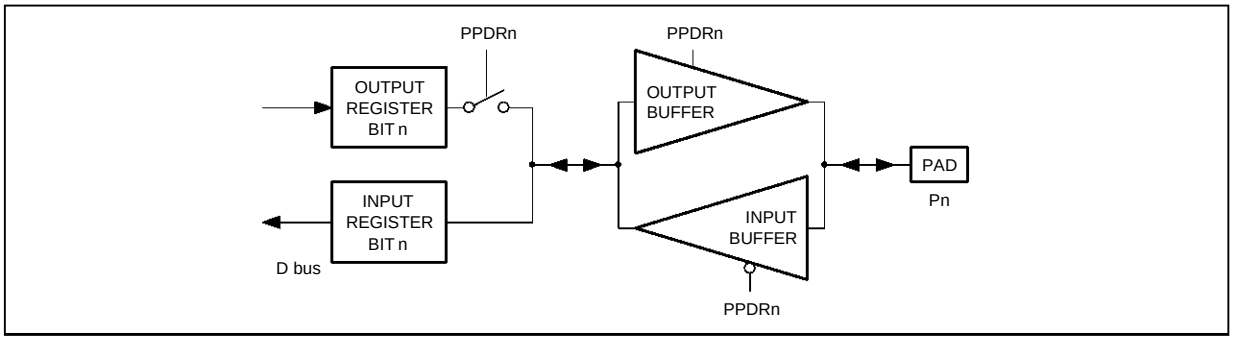


Figure 12

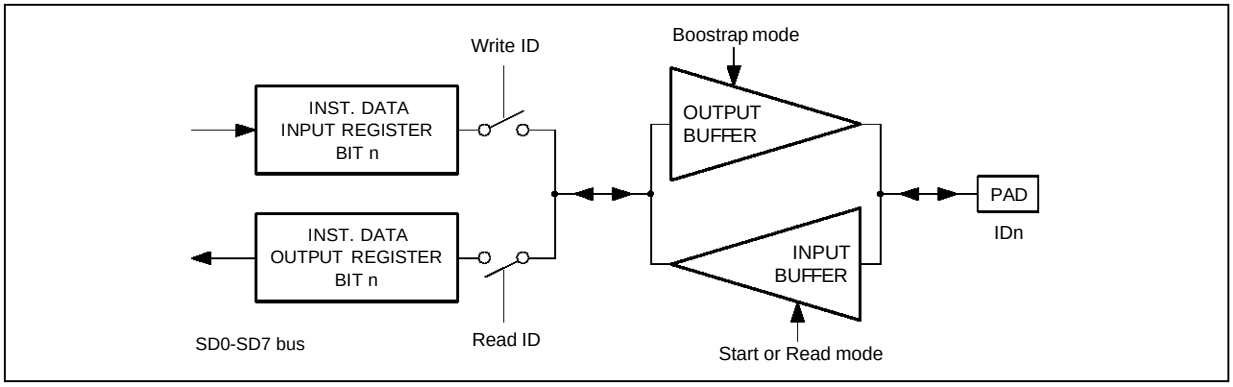
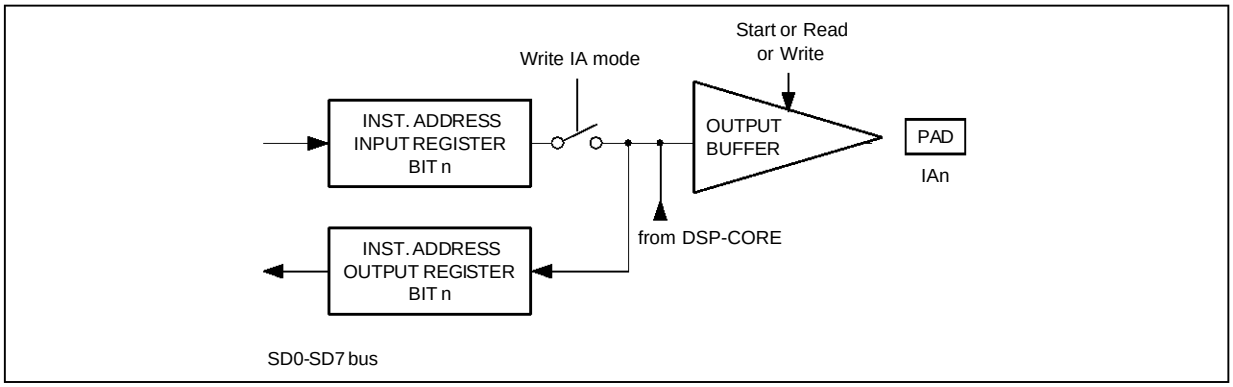


Figure 13



4.1- Bootstrap Command Register

The external program memory read/write is *FORBIDDEN IF* the DSP-CORE is not stopped. The host must halt it first.
Then, it has to write a command word into the 2-bit BOOTCO register (BOOTstrap COmmand) located at the address $70.

The BOOTCO register is defined as follows:

	BC1	BC0	
mode 0	: 0	0	→ Start DSP-CORE (Normal mode)
mode 1	: 0	1	→ Halt DSP-CORE and IA,ID,A,D buses in high Z.
mode 2	: 1	0	→ Halt DSP-CORE and Write external program memory
mode 3	: 1	1	→ Halt DSP-CORE and Read external program memory

This register can be written and read by the host, and doesn't appear in the address space of the DSP-CORE.
The BOOTCO register is LSB-adjusted regarding the system bus, and the bits 2 to 7 are asserted to 0 during a read cycle.

A write cycle into the BOOTCO register (except value: 00) halts the DSP at the end of the NEXT instruction, a NOP instruction is executed while the DSP internal state is kept unchanged.

$\overline{\text{RESET}}$ effects:
- BC0 and BC1 bits are set to 0 by the falling edge of $\overline{\text{RESET}}$;
- The Host can write into the BOOTCO register after 4 CLKOUT cycles delay.
- when $\overline{\text{RESET}}$ remains inactive, the DSP-CORE starts or is halted according to the value of BC0 and BC1.

4.2- Bootstrap Function

During a bootstrap operation:
- the DSP-CORE must be halted
- the Instruction data bus(ID0-ID31) is configured as an output bus
- the instruction data and address buses are fed by the bootstrap logic.

After halting the DSP, the host can begin to write the sets of six bytes from the address $71 to $76.
The bootstrap registers mapping is:

		SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0	
	$70 :							BC1	BC0	BOOTCO reg.
(MSB)	$71 :	IA15							IA8	Inst. Ad.
(LSB)	$72 :	IA7							IA0	Inst. Ad.
(MSB)	$73 :	ID31							ID24	Inst. Dat.
	$74 :	ID23							ID16	Inst. Dat.
	$75 :	ID15							ID8	Inst. Dat.
(LSB)	$76 :	ID7							ID0	Inst. Dat.

A write cycle at the address $72 initializes, at the next CLKOUT rising edge, the output of the instruction address on the IA0-IA15 bus.
The IA0-IA15 bus remains stable and in low impedance until the host writes again at this address.

A write cycle at the address $76 initializes, at the next CLKOUT rising edge, a write into the external program memory addressed by IA0-IA15.

The $\overline{WPM}$ output signal is asserted low and the ID0-ID31 instruction data bus is driven in low impedance during a DSP-CORE machine cycle.

So, the host processor must load:
- the Instruction address first (MSB first)
- the Instruction data last (MSB first).

Then, the host can begin to write a new set of bytes.

4.3 - Read External Program Memory

At the end of the external program memory loading:
- the DSP can restart if the BOOTCO register is loaded with 00,
- or the host can initialize a read of the external program memory by writing the value 11 into the BOOTCO register.

During an external program memory read by the host:
- the DSP-CORE must be halted
- the Instruction data bus(ID0-ID31) is configured as an input bus
- the instruction address bus is fed by the bootstrap logic.

The host writes the 2 bytes of the instruction address ($71 and $72).

The write cycle at the address $72 initializes (at the next CLKOUT rising edge):
- the output of the instruction address on the IA0-IA15 bus
- a read of the external program memory addressed by IA0-IA15 and the data storage into the 4 byte-wide registers located at the addresses $73 to $76.

The IA0-IA15 bus remains stable and in low impedance until the host writes again at the address $72.

In this mode, the $\overline{WPM}$ output signal is always set to 1 (inactive).
The maximum delay between the instruction address write by the host and the availability of the instruction data is TWO DSP-CORE cycles.
The SDTACK output signal is used to synchronize the host read (wait states).

4.4- DSP-Core Halt without Any Access to the Memory

When the host writes the value ”01” into the BOOTCO register, the DSP is halted at the end of the next instruction and a NOP instruction is executed.

The address and data buses (IA0-IA15,ID0-ID31,A0-A15,D0-D15) are asserted in high impedance (in the other modes, they stay in low impedance, except D0-D15) .

This mode allows the host to write and read the byte-wide registers regardless of the bootstrap mechanism. It is the test mode for these registers which does not appear in the address space of the DSP-CORE.

4.5 - DSP-Core Restart

Writing the 00 value into the BOOTCO register allows the DSP to restart from its previous internal state.

APPENDIX 1 - ST18933 REGISTERS SUMMARY AND RESET STATE

$\overline{\text{MEI}}$: Micro Enable Interrupts register

7	6	5	4	3	2	1
Global	Enable	Enable	Enable	Enable	Enable	Enable
Enable	MINT6	MINT5	MINT4	MINT3	MINT2	MINT1

$\overline{\text{MISTATE}}$: Micro Interrupt STATE register

6	5	4	3	2	1	0
State of	State of	State of	State of	State of	State of	State of
MINT6	MINT5	MINT4	MINT3	MINT2	MINT1	MINT0

$\overline{\text{DEI}}$: DSP-CORE Enable Interrupts register

7	6	5	4	3	2	1
Global	Enable	Enable	Enable	Enable	Enable	Enable
	Enable	DINT6	DINT5	DINT4	DINT3	DINT2

$\overline{\text{DISTATE}}$: DSP-CORE Interrupt STATE register

6	5	4	3	2	1	0
State of	State of	State of	State of	State of	State of	State of
DINT6	DINT5	DINT4	DINT3	DINT2	DINT1	DINT0

When RESET occurs, all these registers (MEI, MISTATE, DEI; DISTATE) are cleared to 0.
Then, the interrupts are masked (gobal mask and individual mask) and cleared.

SSI0R1 / SSI1R1 : SSI0 and SSI1 status/control Register 1

9	8	7	6	5	4	3	2	1	0
OE	IE	TA	FPB1	FPB0	BPF1	BPF0	CPHA	FST1	FST0

SSI0R2 / SSI1R2 : SSI0 and SSI1 status/control Register 2

2	1	0
OR	BSR	SE

SSI0R3 / SSI1R3 : SSI0 and SSI1 status/control Register 3

5	4	3	2	1	0	
MSTR	BSR3	BTR2	BTR2	BTR0	MFST	

When $\overline{\text{RESET}}$ occurs, the SSIs status/control registers are cleared to 0.

TCR : SSSI Transmit Control Register

6	5	4	3	2	1	0
$\overline{\text{DXTS}}$	$\overline{\text{DXS1}}$	TCP	TIE	TBPW2	TBPW1	TBPW0

RCR : SSSI Receive Control Register

6	5	4	3	2	1	0
TDERD	CLKI	RCP	RIE	RBPW2	RBPW1	RBPW0

When $\overline{\text{RESET}}$ occurs the SSSI Control Registers are cleared to 0.

PPORT :

3	2	1	0
P3	P2	P1	P0

PPDR : PPort Data Direction

3	2	1	0
PPDR3	PPDR2	PPDR1	PPDR0

When $\overline{\text{RESET}}$ occurs:
- all the PPDR bits are set to 0
- each PPORT bit is configured as an input and cleared to 0.

BOOTCO : BOOTstrap Control register

1	0
BC1	BC0

When $\overline{\text{RESET}}$ occurs, BOOTCO bits are cleared to 0.

I/O MAPPING IN THE DSP-CORE ADDRESS SPACE

The DSP-CORE page number 0 is dedicated to the five communication peripherals :
- Dual Port Ram
- Synchronous Serial I/O 0 : SSI1
- Synchronous Serial I/O 1 : SSI2
- Symplified Synchronous Serial I/O : SSSI
- General Purpose Parallel Port : PPORT

It is 4k words large.

APPENDIX 2 - ST18933 IN THE HOST ADDRESS SPACE

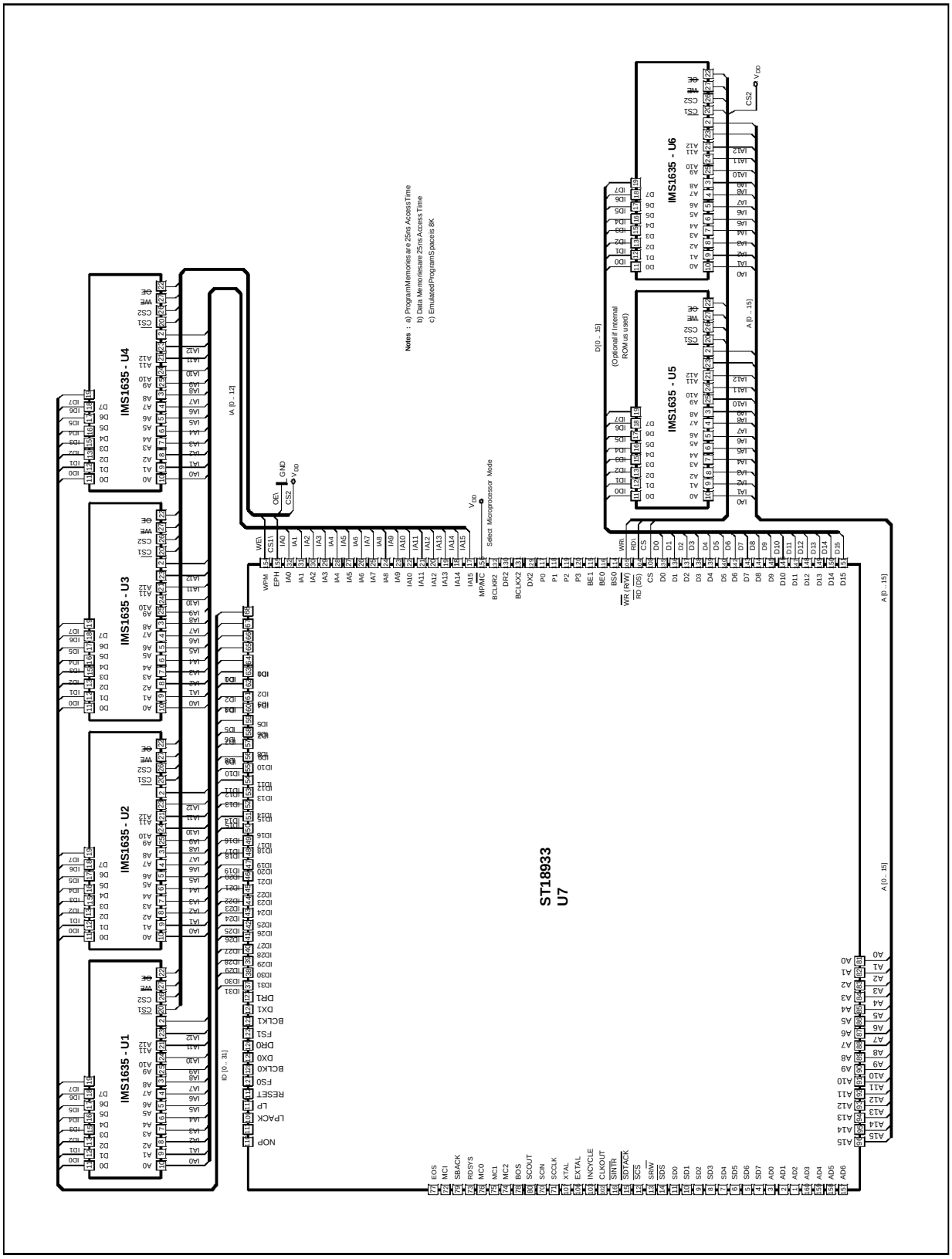
Host Processor space		ST18933 space
$00 : : : $3F	64 x 8-bit DUAL PORT RAM	$0000 : : : $003F
$40	MINT0	$0040
$41	MINT1	$0041
$42	MINT2	$0042
$43	MINT3	$0043
$44	MINT4	$0044
$45	MINT5	$0045
$46	MINT6	$0046
$47	Unused	$0047
$48	DINT0	$0048
$49	DINT1	$0049
$4A	DINT2	$004A
$4B	DINT3	$004B
$4C	DINT4	$004C
$4D	DINT5	$004D
$4E	DINT6	$004E
$4F	MEI	$004F
$50	MISTATE	$0050
$51	DEI	$0051
$52	DISTATE	$0052
$70	BOOTCO	Out of reach
$71	IA15-IA8 reg.	:
$72	IA7 -IA0 reg.	:
$73	ID31-ID24 reg.	:
$74	ID23-ID16 reg.	:
$75	ID15-ID8 reg.	:
$76	ID7 -ID0 reg.	:
$7F*	Unused	$007F : $00FF

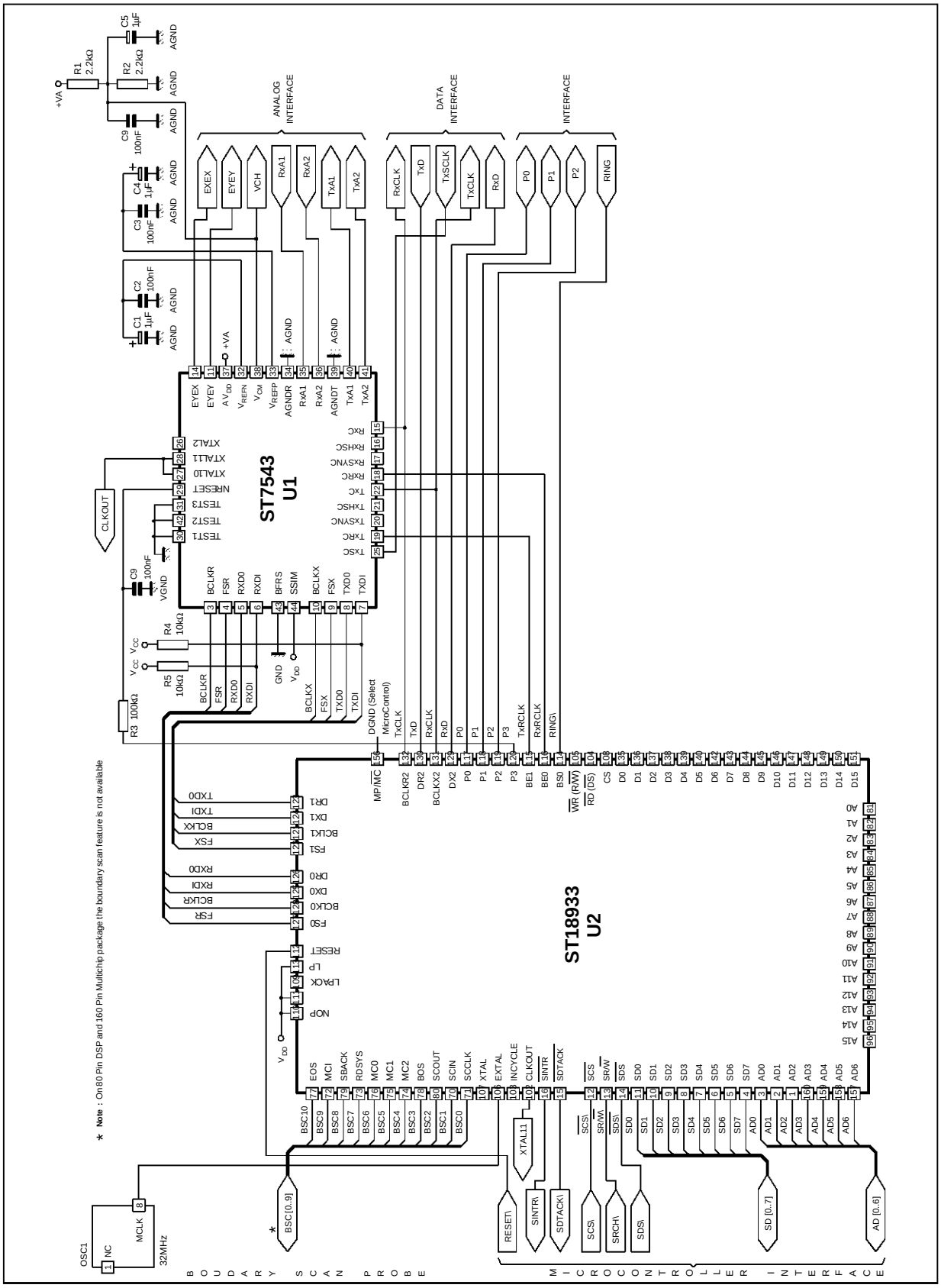
End of MICRO SPACE

APPENDIX 3 - PERIPHERALS MAPPING IN THE ST18933 ADDRESS SPACE

	ST18933 space
SSI0 buffers : transmit if DSP writes receive if DSP reads	$0100 : : $010F
SSI0R1	S0110
SSI0R2	$0111
SSI0R3	$0112
Unused	: : $01FF
SSI1 buffers : transmit if DSP writes receive if DSP reads	$0200 : : $020F
SSI1R1	$0210
SSI1R2	$0211
SSI1R3	$0212
Unused	: : $03FF
SSSI Transmit Reg(SOUT)	$0400
SSSI Receive Reg(SIN)	$0401
SSSI TCR	$0402
SSSI RCR	$0403
Unused	: : $07FF
PPORT	$0800
PPDR	$0801
Unused	$802 : $0FFF

APPENDIX 4 - ST18933 SYSTEM DIAGRAMS




ST7543
U1
ST18933
U2
ANALOG INTERFACE
DATA INTERFACE
INTERFACE
BOUDARY SCAN PROBE
MICROCONTROLLER INTERFACE
* Note : On 80 Pin DSP and 160 Pin Multichip package the boundary scan feature is not available
18933-18.EPS

APPENDIX 5 - PROGRAMMING EXAMPLES

```
"**********************ST18933 PROGRAMMING EXAMPLES**********************"


"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
Convolution routine using complex arithmetic. Coefs and data are 16-bit
complex numbers. Accumulation is on 32-bit real,32-bit imag. Result is stored
in convout, a complex 16-bit rounded number. Number of cycles is (2*firlen)
+ 15 or 7.865 us for dsp with 55ns cycle time and filter length of 64.
This routine can be used for real data by simply changing repeat instruction
to repeat (firlen-2) times ; real sat\
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"

conv:
ini real sat ;;x0 ecd0x\
ini real sat ;;e0 coef\
rini csta cpr\each product is ecd0x*conj(coef)
repeat (firlen-2) times ; cplx sat\
lsr 16 rnd_val.c a\a = ($00008000,$00008000) for rounding purposes
opin ; ldl [x0]+,m; ldr [e0]+,n;\
opin ; ldl [x0]+,m; ldr [e0]+,n;\
begin:
 opin ; ldl [x0]+,m; ldr [e0]+,n;add p a a\accumulate 32-bit real,32-bit imag
end:
opin ;;;add p a a\
opin ;;;add p a a\
svr a convout.y\store rounded output in 16-bit real,16-bit imag
ini real sat\
rini csta nocpr\
rts\

"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
64-tap T/2 cplx equalizer with simultaneous convolution and coef. update.
Cplx coefficients in eram, cplx data in xram addressed in circular mode,
cplx filter output in acc. a, (previous cplx error)*step size in t.
Execution in cplx mode with 2 cplx instruction cycles (4 machine cycles)
per tap.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"

cnvupcof:

ini cplx sat ; ef \
repeat 31 times ; cplx sat\

opin        ;ldl t, m   ; ldr [x0]-,n;clr a     \eqerr*data66
opin        ;           ; ldr [x0]-,n;          \eqerr*data65
opin        ;ldl [x0]-,m; ldr [e0]-,n;addx p r f\data64*coef64,updt coef64
opin        ;ldl [x0]+,m; ldr [e0]+,n;addx p r f\data63*coef63,updt coef63

begin:

opin st f [e0]-;ldl t, m   ; ldr [x0]-,n;add p a a \eqerr*data64,stor coef64
opin st f [e0]-;           ; ldr [x0]-,n;add p a a \eqerr*data63,stor coef63
opin        ;ldl [x0]-,m; ldr [e0]-,n;addx p r f\data62*coef62,updt coef62
opin        ;ldl [x0]+,m; ldr [e0]+,n;addx p r f\data61*coef61,updt coef61

end:

opin st f [e0]-;           ;           ;add p a a \stor coef2
opin st f [e0]-;           ;           ;add p a a \stor coef1,a contains output
rts\
```

Here are 5 LMS examples. Four with convolution and coef. update or update alone for following conditions : Data in XRAM, coefs. in ERAM ; or else all in ERAM. Please note that all data is complex 32 bits (16-bit real and 16-bit imaginary). One can also do the same in the dbpr mode (32-bit real) by simply changing the mode. The 5th example is part of a 64-tap t/2 equalizer using block update (bs = 8) and 64-bit coefs (32-bit real and 32-bit imag parts).

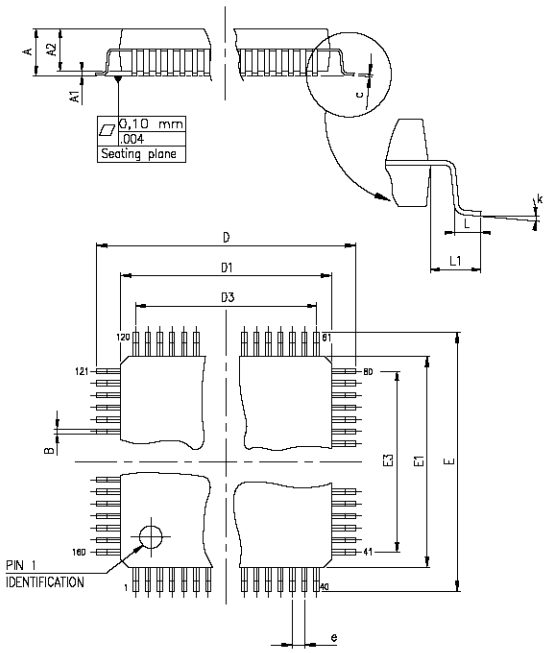
```
"********************** ST1893X BENCHMARKS ****************************

"---------------------------------------------------------------- -------
1.
   20-tap T/2 cplx equalizer with simultaneous convolution and coef. update.
   Cplx coefficients in eram, cplx data in xram addressed in circular mode,
   cplx filter output in acc. a, (cplx error)*step size in t. Execution in
   cplx mode with 2 cplx instruction cycles (4 machine cycles) per tap. The
   data can be stored in eram as well with a transfer of eram to xram before
   execution of the routine adding an additional 1 cplx instruction (2
   machine cycles) per tap.
---------------------------------- ----------------------------------------"
.link

cnvupcof:

ini cplx sat ; ef \
repeat 9 times ; cplx sat\

opin        ;ldl t, m   ; ldr [x0]-,n;clr a     \eqerr*data22
opin        ;           ; ldr [x0]-,n;          \eqerr*data21
opin        ;ldl [x0]-,m; ldr [e0]-,n;addx p r f\data20*coef20,updt coef20
opin        ;ldl [x0]+,m; ldr [e0]+,n;addx p r f\data19*coef19,updt coef19

begin:

opin st f [e0]-;ldl t, m   ; ldr [x0]-,n;add p a a \eqerr*data20,stor coef20
opin st f [e0]-;           ; ldr [x0]-,n;add p a a \eqerr*data19,stor coef19
opin        ;ldl [x0]-,m; ldr [e0]-,n;addx p r f\data18*coef18,updt coef20
opin        ;ldl [x0]+,m; ldr [e0]+,n;addx p r f\data17*coef17,updt coef19

end:

opin st f [e0]-;           ;           ;add p a a \stor coef2
opin st f [e0]-;           ;           ;add p a a \stor coef1
rts\

"---------------------------------------------------------------- ----------
2.
   20-tap cplx T/2 equalizer with coef. update only
   cplx coefficients in eram, cplx data in xram with circular pointer,
   cplx error in multiplier register input register n. Execution in cplx
   mode with 2 cplx instruction cycles (4 machine cycles) per tap.
   The data can be stored in eram as well with a transfer of eram to xram
   before execution of the routine adding an additional 1 cplx instruction
   (2 machine cycles) per tap.
---------------------------------- ----------------------------------------"

upcof:

repeat 20 times ; cplx sat\

opin        ;ldl [x0]+,m;           ;       \d0*error
nop\

begin:

opin        ;ldl [x0]+,m;ldr [e0]   ;add p r a\d1*error,k0+(d0*error
opin st a [e0]+;         ;          ;       \update k0

end:

rts\
```

```
"---------------------------------------------------------------------- ----------
3.
   20-tap T/2 cplx equalizer with simltaneous convolution and coef. update.
   Cplx coefficients in eram, cplx data in eram, cplx filter output
   in acc. b, (cplx error)*step size in [x0]. Assume accu. d contains the
   variable repeat number loaded before call. Execution in cplx mode with 4
   cplx instruction cycles (8 machine cycles) per tap. Assume use of ST18933
   with CROM and ERAM pointers in same page allowing availability of 4 ptrs.
   for ERAM.
---------------------------------------- ----------------------------------------"

cpadapt:

repeat 1 times ; cplx sat\

opin        ;ldl [c0]+,t;; \get data1,assume b accu. already cleared
opdi st d lcr\

begin:

opin        ;ldl [e1]+,m;ldr [x0],n;      \data3*u*err
opin        ;ldl t,m    ;ldr [e0],n;tra r a \data1*coef1,coef1 to a
opin        ;ldl [c0]+ t;        ;add p a a\get data2,accum coef1
opin st a [e0]+;         ;        ;add p b b\update coef1,accum. out

end:

rts\

"---------------------------------------------------------------------- ----------
4.
   20-tap T/2 cplx equalizer with coef. update only.
   Cplx coefficients in eram, cplx data in eram, cplx (error*step size)
   in multiplier input register n. Assume accu. d contains the
   variable repeat number loaded before call. Execution in cplx mode with 3 cplx
   instruction cycles (6 machine cycles) per tap.
---------------------------------------- ----------------------------------------"

cnvupcf1:

repeat 1 times; cplx sat\

opin        ;ldl [e1]+ m;        ;       \data1*u*err
opdi st d lcr\

begin:

opin        ;ldl [e1]+ t;        ;       \data2 to t
opin        ;ldl t,m    ;ldr [e0]  ;add p r a\data2*u*err,acc. coef1
opin st a [e0]+;         ;        ;       \update coef1

end:

rts\

"---------------------------------------------------------------------- ----------
5.
   64-tap T/2 cplx double precision equalizer with coefficient(64-bit)
   update by block method and block size of 8. 64-bit coefficients
   in eram, cplx (32-bit) data in xram, 8 cplx errors (32-bit) in yram.
   Eight coefs are updated per pass. 53*8 = 424 total machine cycles. If
   data is stored in eram, then add 128 machine cycles for loading to xram
   at the start of each baud.

   k(8i + grp) = k(8i + grp) -
       j=7
   u*sum    error(j)*conj(d(8i+grp+2j)
       j=0
```

```
    bsc register contains step size multiplier 2**(-step size)
    x0 points to current cplx data
    e0 points to current cplx coef msb
    e1 points to current cplx coef lsb
    y0 (circular) points to current cplx error
    i  index with 8 values stored in yram
-------------------------------------- -------------------------------"

.asg indxy = 0

get_cof:

ini cplx sat\
opin ;; ldr [e0] ;tra r b\get cplx coef msb
opin ; ldl [e1] t\  get cplx coef lsb
lsr 16 t a\
opin st b t ; ldl t ;;add l a a\64-bit cplx coef to acc. a

repeat 6 times ; cplx sat\

opin ; ldl [y0]+,m ; ldr [x0]+,n;\error*conj(data)
opin ; ldl [y0]+,m ; ldr [x0]+,n;\

begin:
opin ; ldl [y0]+,m ; ldr [x0]+,n;subs p a a\shift by bsc value

end:

opin ;;;subs p a a\
opin ;;;subs p a a\
opin st a [e0];    \store msb of coef
rini csta form,nocpr\
opin st a [e1];    \store lsb of coef
ini real sat\
rini csta noform\
svr x0 t\
opim ldl t,m; ldr 8,n;sub l r a\eqfrd(x+8)
opdi st d x0\update x0 data ptr
svr e0 t\
opim ldl t,m ; ldr 8,n;add l r a\eqfrk(x+8) msb
opdi st d e0\update coef ptrs
svr e1 t\
opim ldl t,m ; ldr 8,n;add l r a\eqfrk(x+8)a lsb
opdi st d e1\
lsl 1 indxy.y a\
svr a indxy.y \
branch nosr get_cof\if 8 coefficients have been updated,finished

.end

"*****************************************************************"
```

PACKAGE MECHANICAL DATA

160 PINS - PLASTIC QUAD FLAT PACK



PMPQF160.EPS

Dimensions	Millimeters			inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			4.07			0.160
A1	0.25			0.010		
A2	3.17	3.42	3.67	0.125	0.135	0.145
B	0.22		0.38	0.008		0.015
C	0.13		0.23	0.005		0.009
D	30.95	31.20	31.45	1.218	1.228	1.238
D1	27.90	28.00	28.10	1.098	1.102	1.106
D3		25.35			0.998	
e		0.65			0.0256	
E	30.95	31.20	31.45	1.218	1.228	1.238
E1	27.90	28.00	28.10	1.098	1.102	1.106
E3		25.35			0.998	
L	0.65	0.80	0.95	0.026	0.031	0.037
L1		1.60			0.063	

PQFP160.TBL