# DSP-93 Workshop

TAPR Annual Meeting

St Louis, Missouri

March 4, 1995

# Workshop Topics

- Hardware Review with Block Diagrams
- TMS320C25
  - Architecture
  - Registers
  - Addressing
  - Instructions
  - Memory
- TLC320C44 - AIO

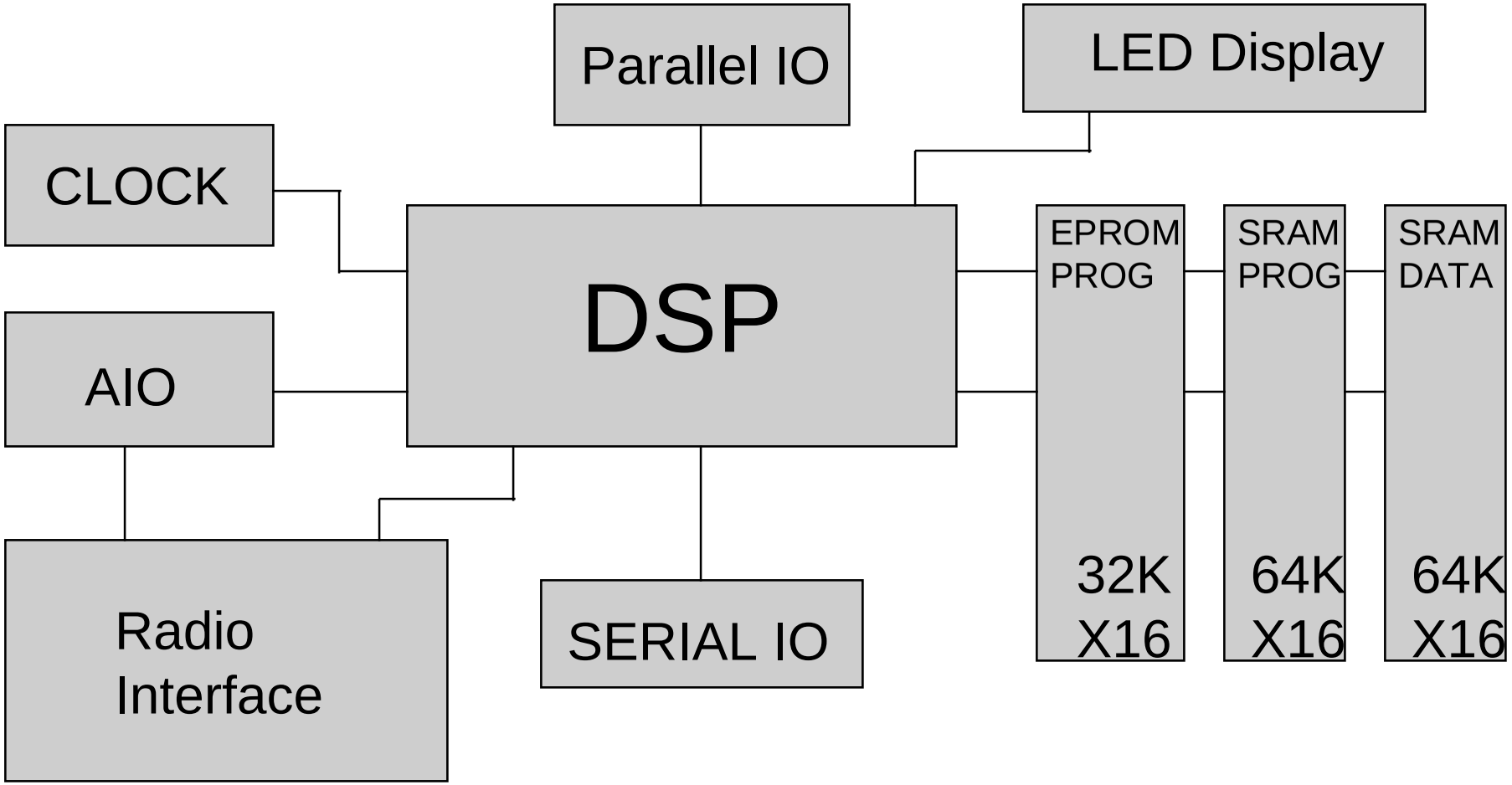
# Workshop Topics

- IO Maps for DSP-93
- Monitor
- Monitor Routines
- TASM Assembler
- DSPLOAD
- Program Examples

# Workshop Topics

- Debug Routine
- D93WE Windows Program Review
- Challenge to the participants
- Review of Information Sources

# Block Diagram



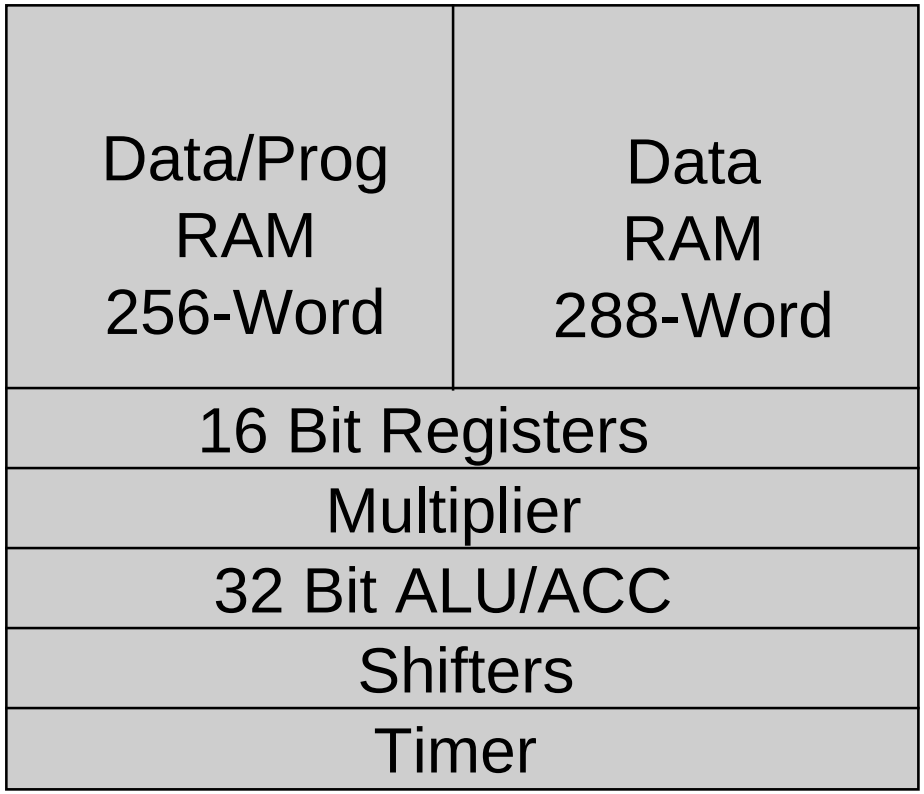
# DSP-93 Hardware

- Radio Interface (2 Ports)
  - Digital
    - PTT
    - Frequency Up and Down
    - Gain (8 Possibilities)
    - Input Source Selection
  - Analog
    - Audio Out
    - Audio In

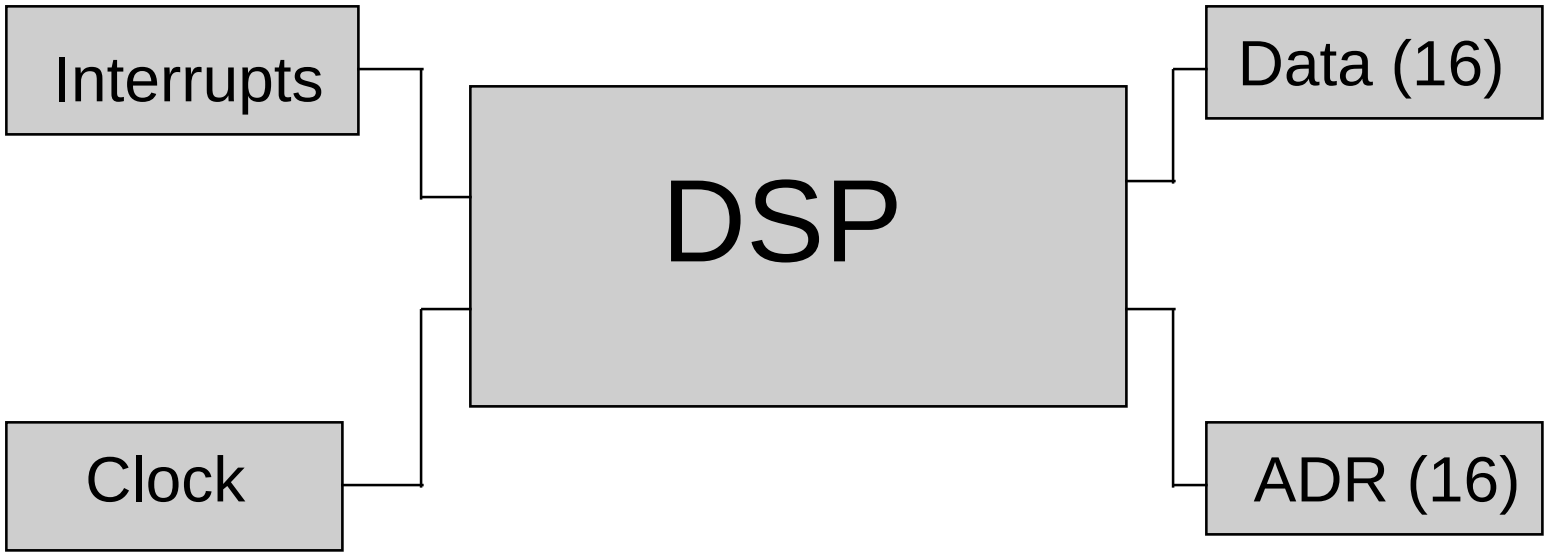
# Architecture

- 16 Bit DSP Chip
- 40 MHz Clock
- Harvard Architecture
  - 64K 16 Bits Program Memory
  - 64K 16 Bits Data Memory

# TMS320C25


Data/Prog
RAM
256-Word
Data
RAM
288-Word
16 Bit Registers
Multiplier
32 Bit ALU/ACC
Shifters
Timer

# TMS320C25



# Key Registers

- Accumulator - 32 Bit Split in two halves
- 8 Auxiliary Registers - AR0 to AR7
  - 16 Bit registers for addressing data, temporary storage, or integer arithmetic
- Interrupt Flag Register - IFR - 5 Bits
- Interrupt Mask Register - IMR - 5 Bits
- Program Counter - PC - 16 Bits

# Key Registers

- Stack - 8 X 16 Bit - Stores PC during interrupts or CALL instructions
- Data Receive Register - DRR
  - » Data coming from the AIO A/D is place here and the processor can be interrupted.
- Data Xmit Register - DXR
  - » Data going to the AIO D/A is place here and then sent. After sending data processor can be interrupted.

# Key Registers

- Status Registers - ST0 & ST1 - 16 Bit
- Timer - TIM - 16 Bit counter

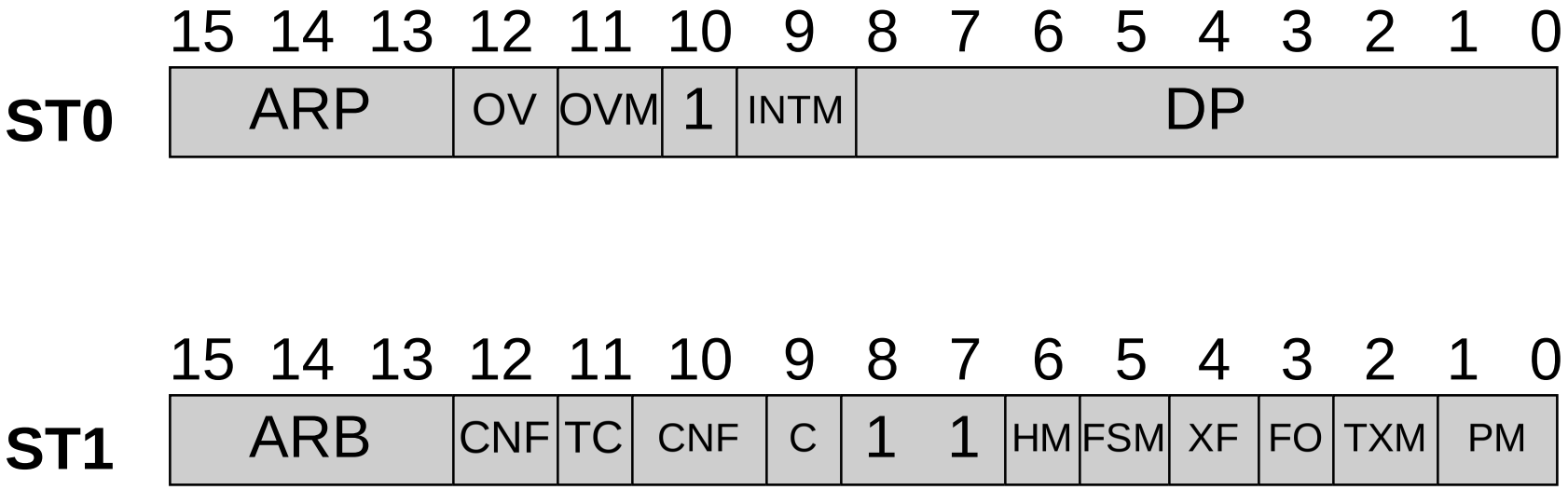
# TMS320C25 AUXILIARY REGISTERS

- Eight Auxiliary Registers
- 16-Bit Unsigned
- Auto Incriment & Decrement
- Auto Indexing using AR0
- Register Pointer and Pointer Buffer
  - ARP, ARB

# Memory Mapped Registers

- DRR and DXR - Serial Port Rec/Xmit
- TIM - Timer
- PRD - Timer Period
- IMR - Interupt Mask
- GREG - Global Memory - For Shared Memory, not used in DSP-93.

# Status Registers




**SST SST1 LST LST1**

# Addressing Modes

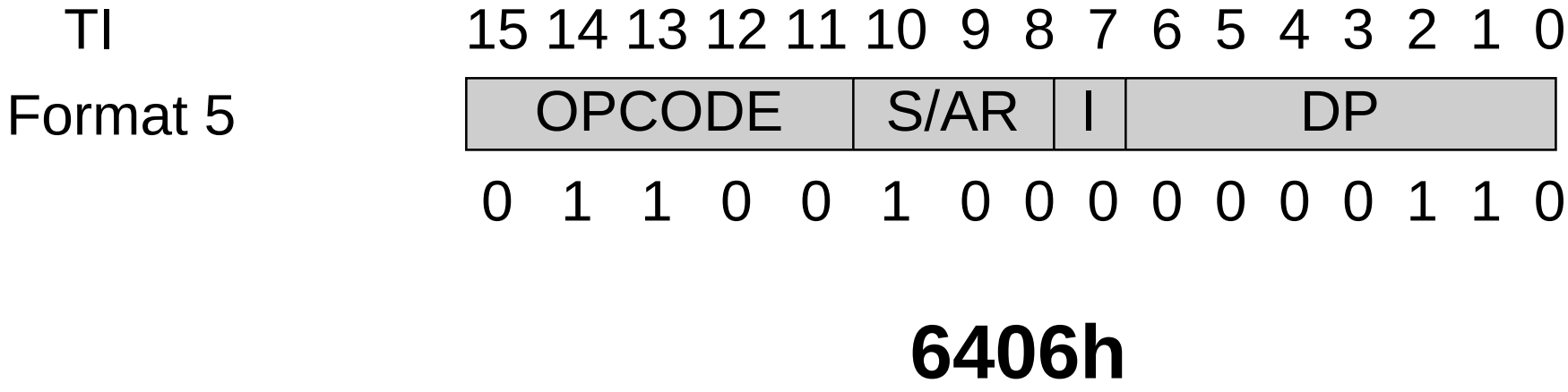
- Immediate
  - ADDK 5
  - ADLK 1F33h
- Direct
  - Uses the DP (9 bits) portion of current ARP
- Indirect Addressing
  - Uses currect ARP as a pointer

# Indirect Addressing

- Uses the LARP instruction to establish an ARP register pointer in ST0.
- Initial memory location must be placed in the selected ARP register.

# Direct Addressing Example

```
        LDPK   8h       ; point to data page 8 - 0400h
⇨       SACL   06h,4    ; Store ACCL at location 0460h
```



# Indirect Addressing

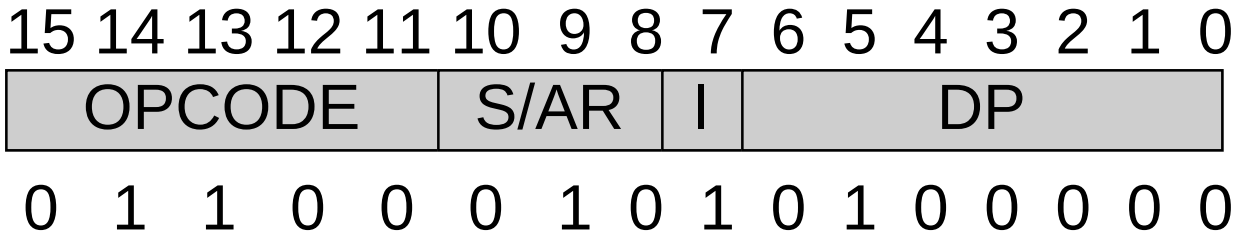
- `SACL * [,Next ARP]` `No Chg current AR`
- `SACL *+ [,Next ARP]` `Inc current AR`
- `SACL *- [,Next ARP]` `Dec current AR`
- `SACL *0+ [,Next ARP]` `AR0 + current AR`
- `SACL *0- [,Next ARP]` `AR0 - current AR`
- `SACL *BR0+ [,Next ARP]` `AR0 + current AR (note)`
- `SACL *BR0- [,Next ARP]` `AR0 - current AR (note)`

Note: Uses reverse carry propagation.

# Indirect Addressing Example

```
        LARP    2       ; Set ST0 to point to AR2
        LARK    0460h   ; Load AR2 with 460h
=>      SACL    *+,3    ; Store ACCL at location 0460h
                        ; Incriment AR2 to 461h and change
                        ; ST0 to point to AR3
```

TI Format 5




**62A0h**

# TMS320C25 Instructions

LOAD STORE LOGICAL

ALU - SHIFT

ARITHMETIC

ADD SUB NEG ABS

MPY SQRA

Memory Move Branch Control

# Multiply Instructions I

- MPY - Multiply T-Reg w/data
- MPYK - MPY T-Reg w/13-bit constant
- MPYU - MPY T-Reg w/unsigned data
- MPYA - MPY T-Reg w/data & add past P-Reg to ACC
- MPYS - MPY T-Reg w/data & subtract past P-Reg from ACC

# P-Register - 16 Bits

- PAC Load ACC with P-Reg
- APAC Add P-Reg to ACC
- SPAC Subtract P-Reg from ACC
- LPH Load P-Reg high
- SPH Store P-Reg high
- SPL Store P-Reg low

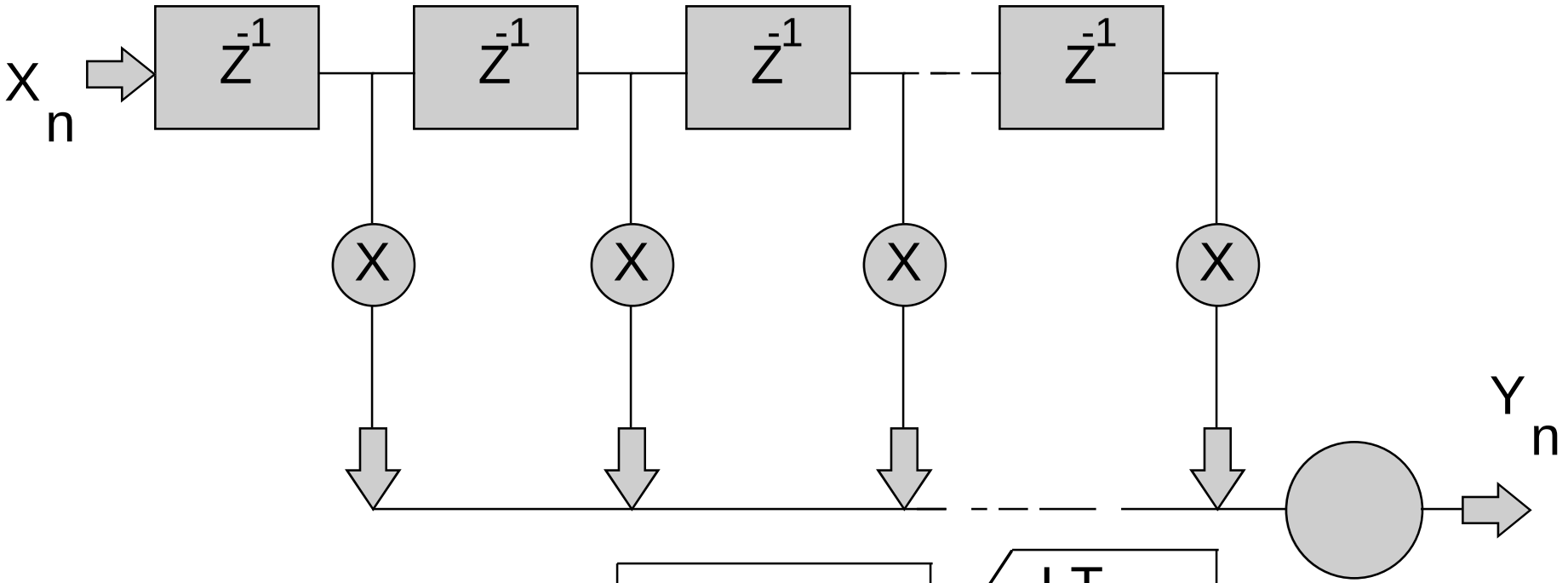
# T-Register - 16 Bits

- LT — Load T-Reg
- LTP — Load T-Reg & move P-Reg to ACC
- LTA — Load T-Reg & add P-Reg to ACC
- LTS — Load T-Reg & subtract P-Reg from ACC
- LTD — Load T-Reg, add P-Reg to ACC, & move data to next memory location.

# Multiply Instructions II

- MAC - MPY data memory * program memory & add past P-Reg to ACC
- MACD - MPY data memory * program memory, add past P-Reg to ACC, & move data memory
- SQRA - Square data memory value & add past P-Reg to ACC
- SQRS - Square data memory value & sub past P-Reg from ACC

# FIR Filter




$$Y_n = \sum_{k=0}^{N} b_k X_{(n-k)}$$

RPTK 49
MACD

LT
DMOV
APAC
MPY

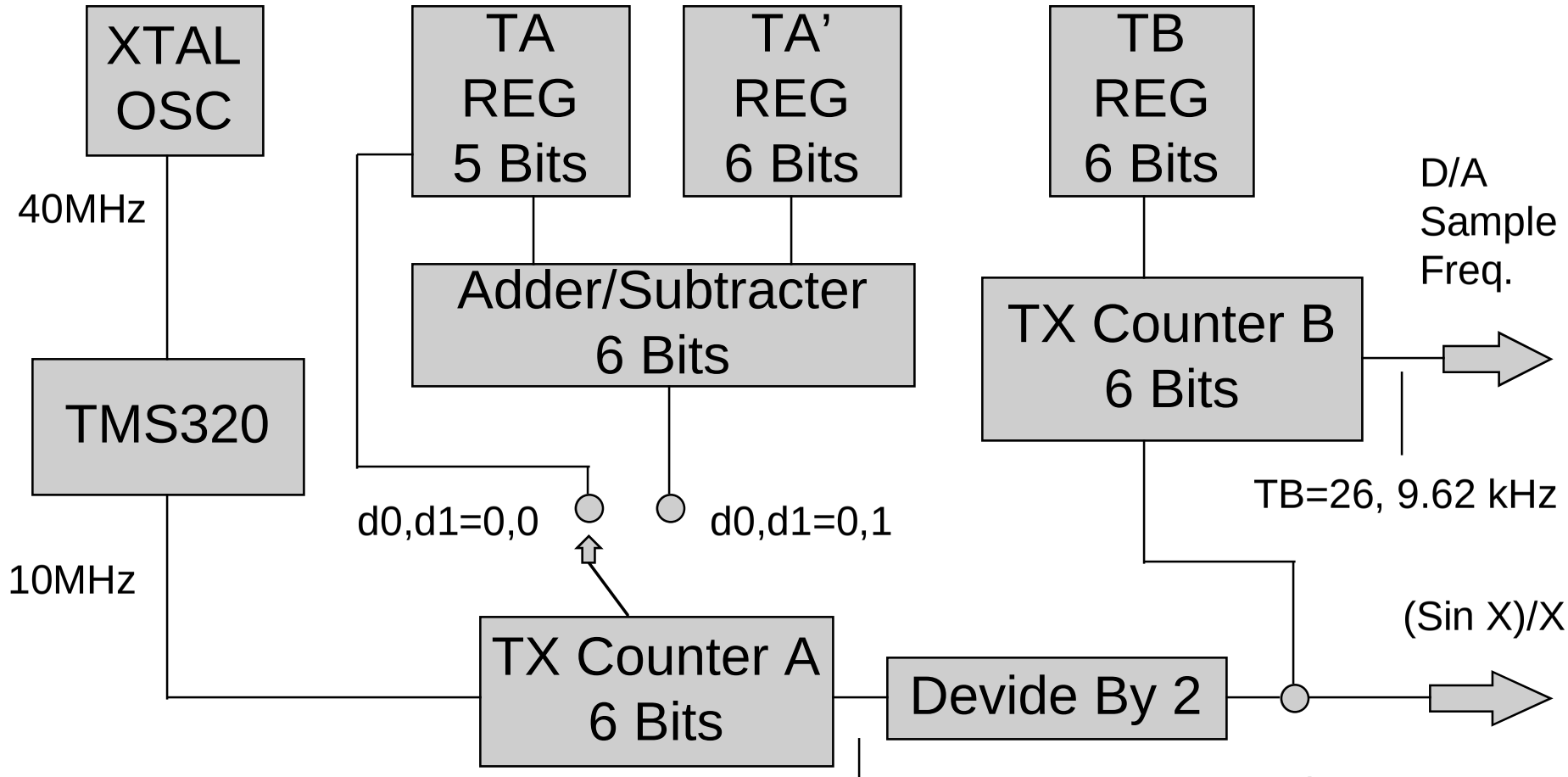
# TLC320C44

- ADC and DAC
  - 14 Bit Dynamic Range
  - 19.2 K Samples per Second (44K Possible)
- Switched - Capacitor Filter
  - Antialiasing Input
  - Output - Reconstruction
- 5 MHz Serial Interface to DSP Chip
  -

# TLC320C44

- Review Block diagram on page 5-71 of TI linear data book.

# TLC320C44 Internal Timing
# TA, TB, RA, RB Registers



# On-Chip Memory

- DSP-93 Operating in Microprocessor Mode - MP/$\overline{MC}$=1
- 544 Words On Chip as Data Memory
  - 060h to 07Fh
- 256 Words Configurable
  - CNFD Inst. - Data Memory - 0200h to 02FFh
  - CNFP Inst. - Prog Memory - FF00h to FFFFh

# Port Assignments

| | |
|---|---|
| PORT 01h | UART WRITE |
| PORT 02h | HIGH SPEED CLK & H_AIO |
| PORT 03h | UART CONTROL REGISTER (WDT STROKE) |
| PORT 04h | H_AIO SELECT OUTPUT |
| PORT 05h | UART READ (READ ONLY) |
| PORT 06h | TNC OUTPUT |
| PORT 07h | RADIO PORT SELECT AND GAIN & AIO ENABLE |
| PORT 0Ah | TNC INPUT (READ ONLY) |
| PORT 0Bh | RADIO PORT DIGITAL CONTROL |
| PORT 0Ch | Network Interface |
| PORT 0Dh | Network Interface |

# IO Port Usage

PORT 07h RADIO PORT SELECT AND GAIN & AIO ENABLE

| CODE D2 D1 D0 | CONNECTION | NAME | DESCRIPTION |
|---|---|---|---|
| 0 0 0 | I/O201 PIN 3 | AUX IN | AUXILIARY AUDIO IN RADIO 1 |
| 0 0 1 | I/O202 PIN 3 | AUX IN | AUXILIARY AUDIO IN RADIO 2 |
| 0 1 0 | Jx03 PIN 14 | AUX #2 | AUDIO BUSS #2 |
| 0 1 1 | Jx03 PIN 12 | AUX #1 | AUDIO BUSS #1 |
| 1 0 0 | I/O202 PIN 5 | XMIT 2 | TRANSMIT MONITOR RADIO 2 |
| 1 0 1 | I/O201 PIN 8 | REC IN | RECEIVE AUDIO RADIO 1 |
| 1 1 0 | I/O201 PIN 5 | XMIT 1 | TRANSMIT MONITOR RADIO 1 |
| 1 1 1 | I/O202 PIN 8 | REC IN | RECEIVE AUDIO RADIO |

# IO Port Usage

PORT 07h RADIO PORT SELECT AND GAIN & AIO ENABLE

| CODE D5 D4 D3 | CONNECTION NAME | DESCRIPTION |
|---|---|---|
| 0 0 0 | FEEDBACK | GAIN= Use MEASGAIN to determine your |
| 0 0 1 | FDBK R201 | GAIN= units gain. |
| 0 1 0 | FDBK R204 | GAIN= |
| 0 1 1 | FDBK R205 | GAIN= |
| 1 0 0 | FDBK R206 | GAIN= |
| 1 0 1 | FDBK R207 | GAIN= |
| 1 1 0 | FDBK R208 | GAIN= |
| 1 1 1 | FDBK R203 | GAIN=MAX |

# Monitor Menu

*?

DSP-93 TAPR / AMSAT REV 2.16

Copyright <c> 1995 by Tucson Amateur Packet Radio Corporation.

A-AR REGISTERS
D-DUMP MEMORY
F-FILL MEMORY
G-FLIP & RUN PROGRAM @ 1008h
H-INTEL LOADER HIGH BITS
J-JUMP TO XXXX & RUN
L-INTEL LOADER  LOW BITS
M-MODIFY WORD
P-FIRMWARE PROGRAMS
R-RESET
S-SHOW WORD
T-TEST

# On Chip Programs

```
PROGRAM #(1-X    ?)=
?
1-SCOPE                    C-PSK_93SK        N-TESTPORT
2-SPECTRUM ANALYZER        D-PSK_93XI        O-TESTRAMP
3-FUNCTION GENERATOR       E-PSK_93XK        P-TPRSP1
4-CW_93D                   F-PSK_93AI        Q-TPRSP2
5-HFU_93X                  G-PSK_93AK        R-FSKP1
6-HF_93A                   H-PSK_93T2        S-FSKP2
7-HF_93AE                  I-PSK_93P2        T-FSKISIP1
8-MEMTEST                  J-APT_93B         U-FSKISIP2
9-PKTP1                    K-TESTANIN        V-MEASGATN
A-PKTP2                    L-TESTBUZZ        W-MEASGAIP
B-PSK_93SI                 M-TESTLEDS        X-SPEEDTST
```

# Monitor Subroutines

```
DEBUG    .EQU   0400h      ; Start of N5EG Debue
                             Routine
GETVAL16 .EQU   0882h      ; Get data from SIO, result
                           in 0003 of Current Page
GET4HEX  .EQU   0402h      ; Get four HEX digits.
GET2HEX  .EQU   0404h      ; Get  two HEX digits.
GETCHAR  .EQU   0406h      ; Get  one HEX digit.
HEXOUT   .EQU   0408h      ; Sends data in accumulator
                           to SIO
INBIT    .EQU   040Ah      ; Rec data from SIO to
                           AR(ARP)
OUTBIT   .EQU   040Ch      ; Sends Lower 8 bits of
                           AR(ARP) to SIO (Uses 0060)
```

# Monitor Subroutines

```
OUTBIT2 .EQU    040Eh     ; Sends lower 8 bits of
                            0062 to SIO (Uses 0060)
PRVAL08 .EQU    0410h     ; Print value at 0006 to SIO
PRVAL16 .EQU    0412h     ; Print value at 0006 to SIO
RESET   .EQU    0414h     ; Reset the DSP-93
SD_CRLF .EQU    0416h     ; Send CR and LF to SIO
SD_STR  .EQU    0418h     ; Print string at (AR0).  C
                            style strings used
```

# Monitor Subroutines

```
SP1     .EQU    041Ah       ; Send 1 space to SIO
SP2     .EQU    041Ch       ;      2
SP3     .EQU    041Eh       ;      3
SP4     .EQU    0420h       ;      4
SP5     .EQU    0422h       ;      5
SP6     .EQU    0424h       ;      6
SP7     .EQU    0426h       ;      7
SP8     .EQU    0428h       ;      8
SP9     .EQU    042Ah       ;      9
SP10    .EQU    042Ch       ;      10
```

# Monitor Subroutines

```
MWAIT_A   .EQU    042Eh  ; Entry for about N * 1 mS
                         ; wait.
                         ; N is the value in AR2
MWAIT1    .EQU    0430h  ; Entry for a 32 mS wait.

PM_BLK_MV .EQU    0432h  ; Moves a block of code
                         ; PROG -> DATA
MP_BLK_MV .EQU    0434h  ; Moves a block of code
                         ; DATA -> PROG
```

# Example Program

```
.title  "Example Minimum Program"
; Example of a minimum Assembly Program for
    the DSP93
; Bob Stricklin N5BRG
; 1/3/94

#include "PORTS.INC"
#include "MONITOR.INC"
#include "REGS.INC"
```

# Example Program

```
.ORG    1000h

B       TINT    ;  Timer interrupt service routine
B       RINT    ;  SIO receive interrupt service
                   routine
B       XINT    ;  SIO transmit interrupt service
                   routine
B       TRAP    ; Software trap interrupt service
                   routine
```

# Example Program

```
INIT
    LDPK    8h              ; point to data page eight (8)
    LARP    AR0             ; Establish a pointer to a AR
                              register.
    LRLK    AR0,2000h       ; Set up a known memory
                              area in pointer
```

# Example Program

```
MAINLOOP
; Your main Program body goes here.
RPTK     S_END-STR1
   BLKP     STR1,*+
   LRLK     AR0,2000h
   CALL     SD_STR
   CALL     SD_CRLF
   B  S_END
```

# Example Program

```
; ---------------------------------------------------------------
STR1 .TEXT    "E x a m p l e   P r o g r a m"
        .WORD    00
S_END
```

# Example Program

```
TINT
; Your program code goes here.
    RET

RINT
; Your program code goes here.
     RET
```

# Example Program

```
XINT
; Your program code goes here.
    RET

TRAP
; Your program code goes here.
   RET
```

# Example Program

```
S_END

.END                ; end of program
```

# DSPLOAD Program

- Load Intel Hex Files into DSP-93
- Configures PC Port per CFG file.
- Resets DSP-93 - LED 1 Flashes
- Checks DSP-93 For response (*)
- Downloads Hi bytes and then Low bytes
- Issues a ‘G’ command.

# DSPLOAD CFG

```
1 Port for serial IO; 1=COM1, 2=COM2,
                      3=COM3, 4=COM4
19200 Baud rate for all IO; 1200, 2400,
                            4800, 9600,
                          or 19200
8 Word length of data word; 5, 6, 7, or 8
1 Number of Stop bits; 1, or 2
0 Type of Parity; 0=None, 1=Odd, 2=Even
```

# TASM Assembler BAT File

TASM -3225 -lal -g0 PROGRAM.ASM

DSPLOAD PROGRAM

-3225 Says Source is TMS**32**0C**25**

-lal Creates a listing showing all labels in the long form

-g0 Sets Object file format to Intel Hex

# Workshop Disk Contents

- TASM Assembler
- Sample Programs
- TI  BBS File Listing
- DSP-93 Include Files
- Workshop Presentation
- DSP-93 Listserver Archives DEC-FEB