1. Explain Functional Architecture of TMS320C54XX Processor, with a Block Diagram.

Sol:

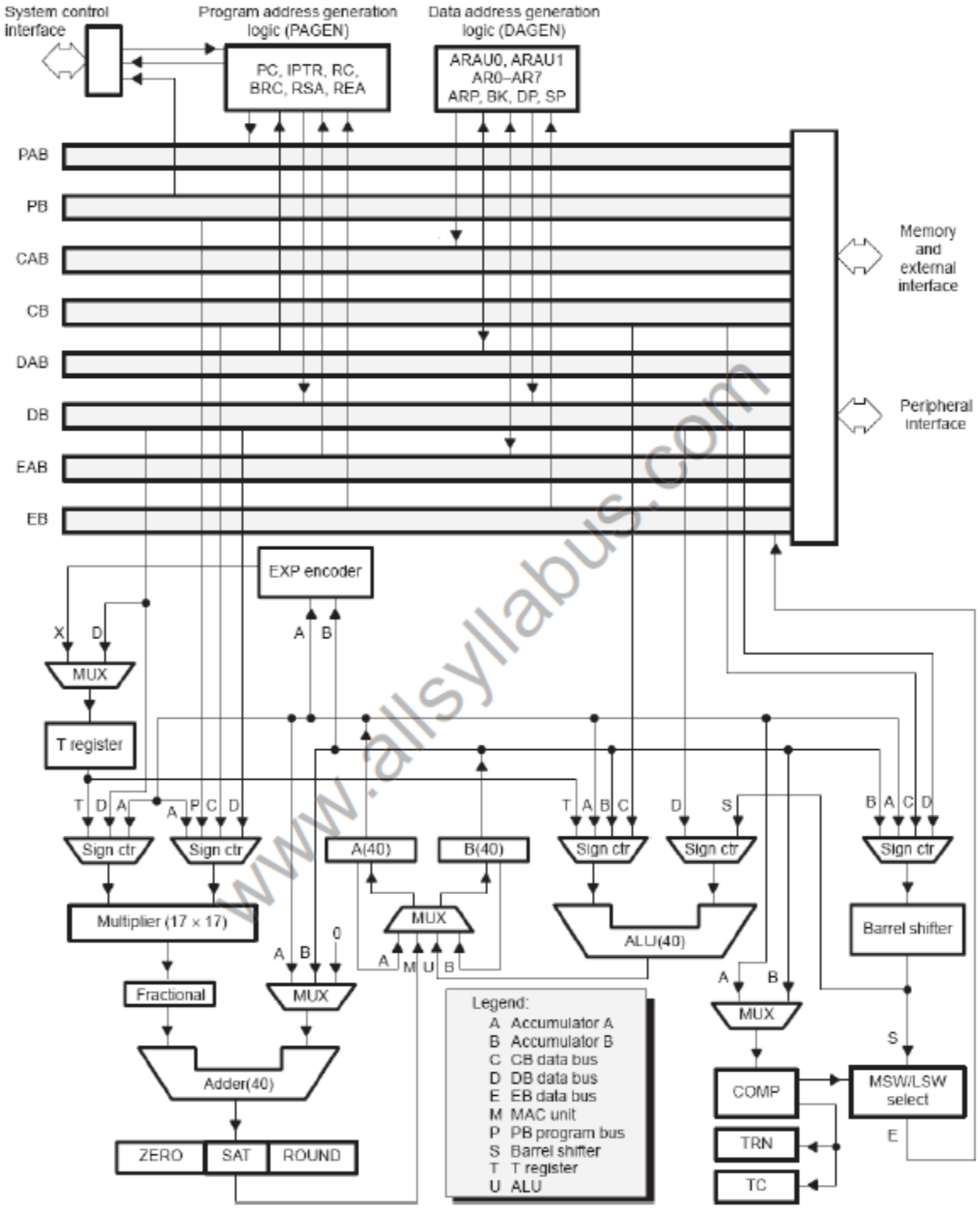


Figure 3.1.Functional architecture for TMS320C54xx processors.

3.3.1 Bus Structure:

The performance of a processor gets enhanced with the provision of multiple buses to provide simultaneous access to various parts of memory or peripherals. The 54xx architecture is built around four pairs of 16-bit buses with each pair consisting of an address bus and a data bus. As shown in Figure 3.1, these are The program bus pair (**PAB, PB**); which carries the instruction code from the program memory. Three data bus pairs (**CAB, CB; DAB, DB**; and **EAB, EB**); which interconnected the various units within the CPU. In Addition the pair CAB, CB and DAB, DB are used to read from the data memory, while The pair **EAB, EB**; carries the data to be written to the memory. The '54xx can generate up to two data-memory addresses per cycle using the two auxiliary register arithmetic unit (ARAU0 and ARAU1) in the DAGEN block. This enables accessing two operands simultaneously.

3.3.2 Central Processing Unit (CPU):

The '54xx CPU is common to all the '54xx devices. The '54xx CPU contains a 40-bit arithmetic logic unit (**ALU**); two 40-bit accumulators (**A** and **B**); a barrel shifter; a

17 x 17-bit multiplier; a 40-bit adder; a compare, select and store unit (**CSSU**); an exponent encoder(**EXP**); a data address generation unit (**DAGEN**); and a program address generation unit (**PAGEN**).

The ALU performs 2's complement arithmetic operations and bit-level Boolean operations on 16, 32, and 40-bit words. It can also function as two separate 16-bit ALUs

and perform two 16-bit operations simultaneously. Figure 3.2 show the functional diagram of the ALU of the TMS320C54xx family of devices.

Accumulators A and B store the output from the ALU or the multiplier/adder block and provide a second input to the ALU. Each accumulators is divided into three parts: guards bits (bits 39-32), high-order word (bits-31-16), and low-order word (bits 15- 0), which can be stored and retrieved individually. Each accumulator is memory-mapped and partitioned. It can be configured as the destination registers. The guard bits are used as a head margin for computations. [5M]

2 (a). Discuss briefly about the Direct and Indirect addressing modes of TMS320C54XX Processor

Sol:

Base address + 7 bits of value contained in instruction = 16 bit address. A page of 128 locations can be accessed without change in DP or SP. Compiler mode bit (CPL) in ST1 register is used. If CPL =0, selects DP ; CPL = 1, selects SP.

When SP is used instead of DP, the effective address is computed by adding the 7-bit offset to SP

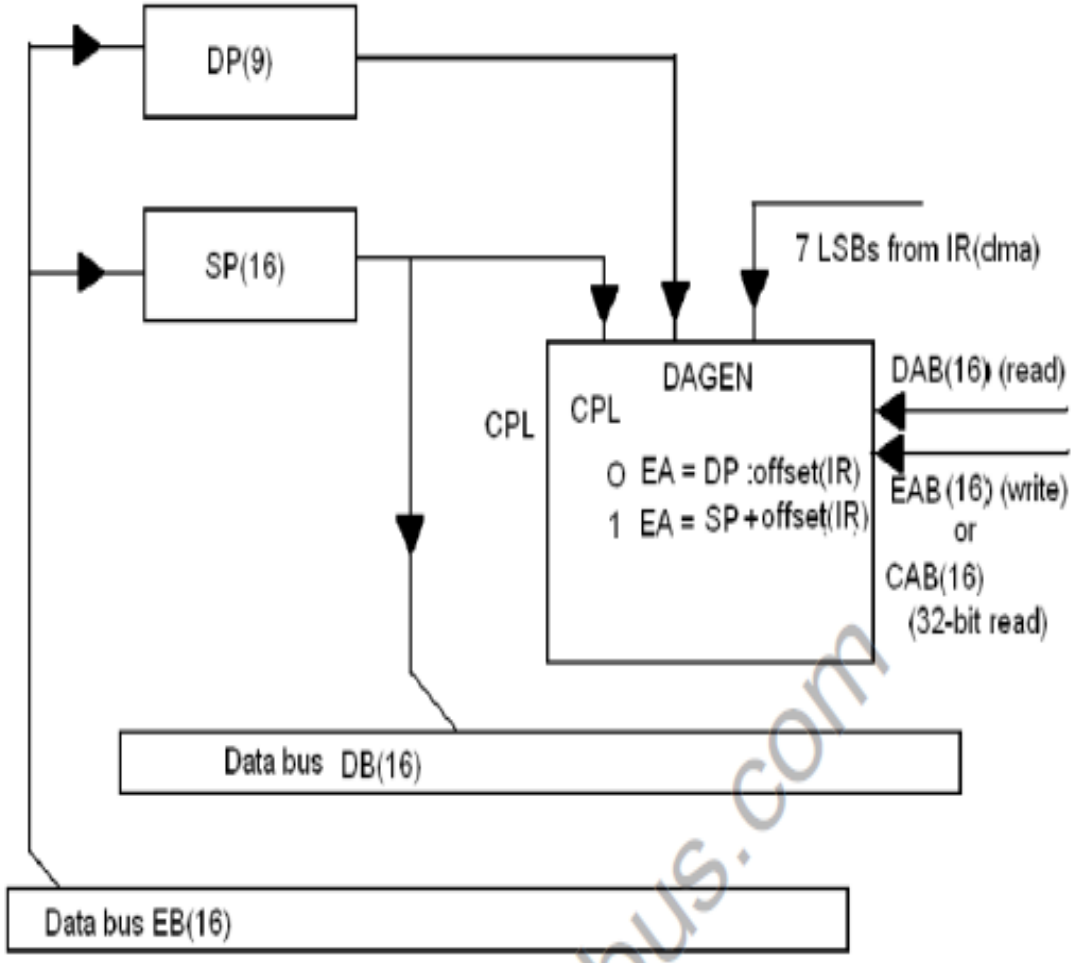


Figure 3.7 Block diagram of the direct addressing mode for TMS320C54xx Processors.

Indirect addressing mode

- MS320C54xx have 8, 16 bit auxiliary register (AR0 – AR 7). Two auxiliary register arithmetic units (ARAU0 & ARAU1)
- Used to access memory location in fixed step size. AR0 register is used for indexed and bit reverse addressing modes.
- For single operand addressing

  MOD → type of indirect addressing

  ARF → AR used for addressing
- ARP depends on (CMPT) bit in ST1

- CMPT = 0, Standard mode, ARP set to zero

CMPT = 1, Compatibility mode, Particularly AR selected by ARP

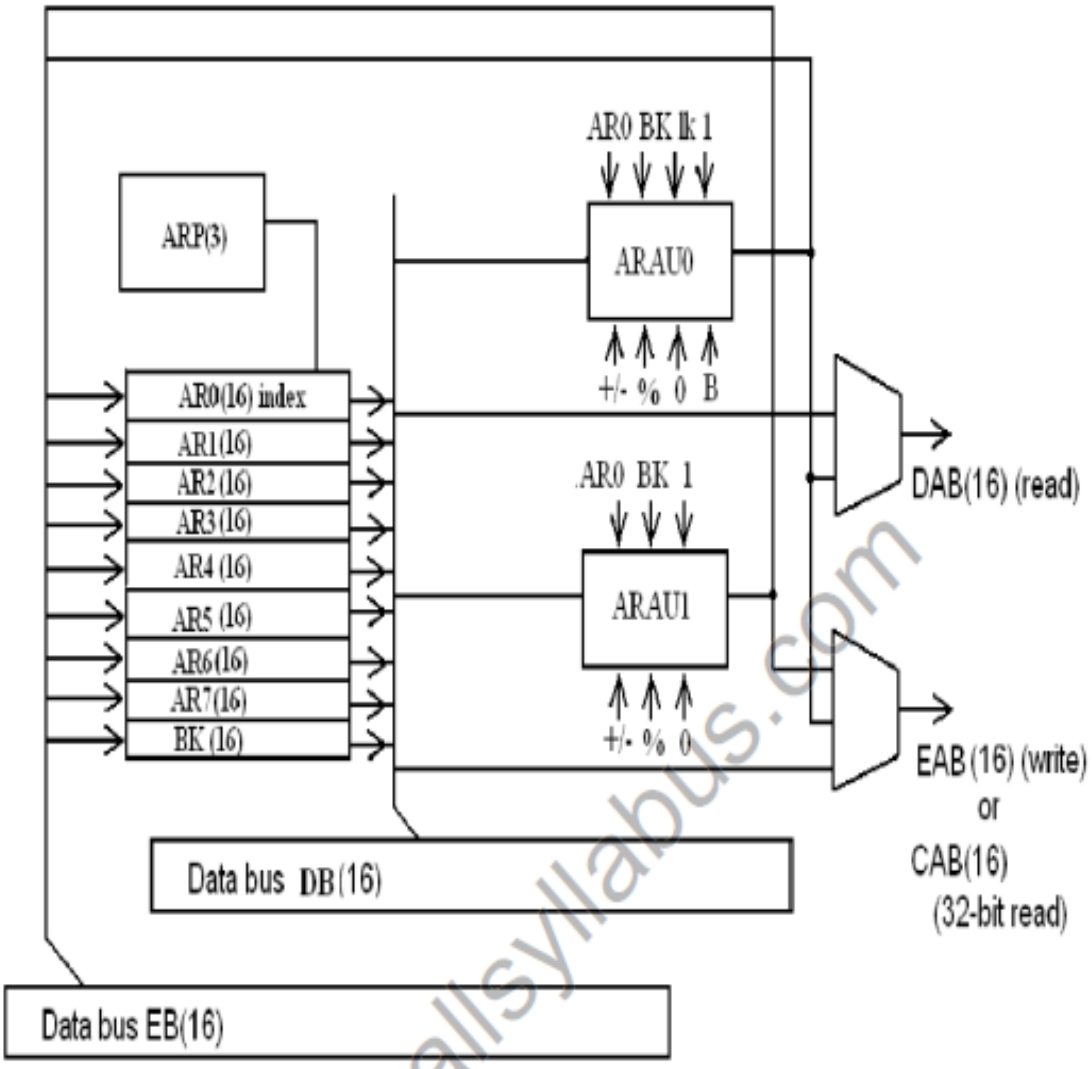


Figure 3.8 Block diagram of the indirect addressing mode for TMS320C54xx Processors.

2 (b). Assuming the current content of AR3 to be 220h, what will be its contents after each of the following TMS320C54xx addressing modes is used? Assume that the contents of AR0 are 10h.

a. *AR3+0 b. *AR3-0 c. *AR3+ d. *AR3 e. *AR3 - f. *+AR3 (40h)

Sol:

a) 230h b) 210h c) 221h d) 220h e) 21Fh f) 260h

3. Write an assembly language program of TMS320C54XX Processors to compute the sum of three product terms given by the equation, $y(n) = h(0)\,x(n) + h(1)\,x(n-1) + h(2)\,x(n-2)$ with usual notations. Use MAC Instruction and Indirect Addressing mode.

Sol:

```
.global _c_int00

.data

.bss x, 3

.bss y, 2

h .int 10, 20, 30

.text

_c_int00:

SSBX SXM ;    Select sign extension mode

STM #x, AR2 ;          Initialize AR2 to point to x(n)

STM #h, AR3 ;          Initialize AR3 to point to h(0)

LD #0H, A ;            Initialize result in A = 0

RPT #2 ;               Repeat the next operation 3 times

MAC *AR2+,*AR3+,A ;          y(n) computed

STM #y, AR2 ;          Select the page for y(n)

STL A, *AR2+ ;         Save the low part of y(n)

STL A, *AR2+ ;         Save the high part of y(n)

NOP ;                        No operation

.end
```

4 (a). Explain the 6 level pipeline operation of TMS320C54XX

Sol:

The CPU of '54xx devices have a six-level-deep instruction pipeline. The six stages of the pipeline are independent of each other. This allows overlapping execution of

instructions. During any given cycle, up to six different instructions can be active, each at a different stage of processing. The six levels of the pipeline structure are program prefetch, program fetch, decode, access, read and execute.

1 During program prefetch, the program address bus, PAB, is loaded with the address of the next instruction to be fetched.
2 In the fetch phase, an instruction word is fetched from the program bus, PB, and loaded into the instruction register, IR. These two phases from the instruction fetch sequence.
3 During the decode stage, the contents of the instruction register, IR are decoded to determine the type of memory access operation and the control signals required for the data-address generation unit and the CPU.
4 The access phase outputs the read operand's on the data address bus, DAB. If a second operand is required, the other data address bus, CAB, also loaded with an appropriate address. Auxiliary registers in indirect addressing mode and the stack pointer (SP) are also updated.
5 In the read phase the data operand(s), if any, are read from the data buses, DB and CB. This phase completes the two-phase read process and starts the two-phase write processes. The data address of the write operand, if any, is loaded into the data write address bus, EAB.
6 The execute phase writes the data using the data write bus, EB, and completes the operand write sequence. The instruction is executed in this phase.

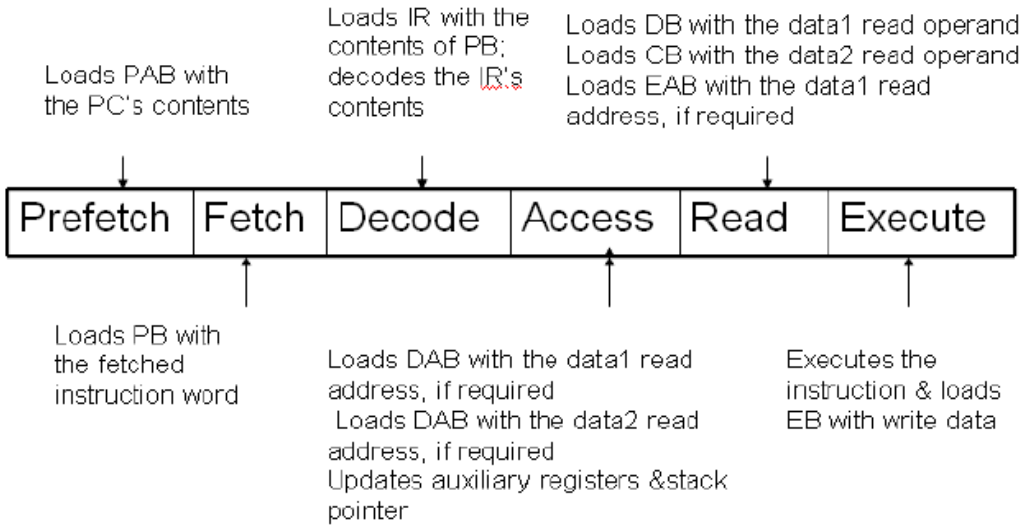


Figure 4.4. Pipeline operation of TMS320C54xx Processors

4 (b). By means of a figure explain the pipeline operation of the following sequence of instruction if the initial values of AR1,AR3,A are 104,101, 2 and the values stored in the memory locations 101,102,103,104 are 4,6,8,12.Also provide the values of registers AR3,AR1,T and accumulator after completion of each cycle.

ADD *AR3+,A

LD*AR1+,T

MPY*AR3+,B

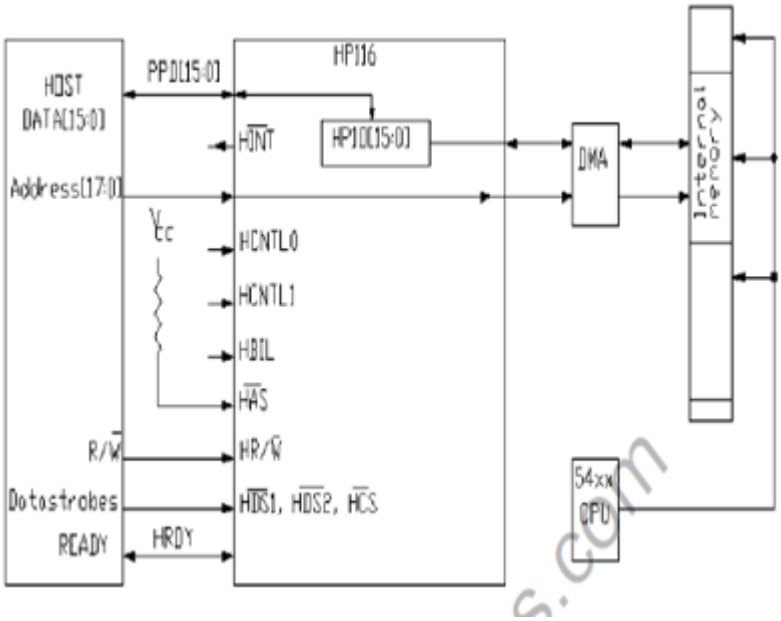
ADD B,A

Sol:

Cycles	Prefetch	Fetch	Decode	Access	Read	Exe& Write	AR3	AR1	A	T
1	ADD						101	104	2	X
2	LD	ADD					101	104	2	X
3	MPY	LD	ADD				101	104	2	X
4	ADD	MPY	LD	ADD			102	104	2	X
5		ADD	MPY	LD	ADD		102	105	2	X
6			ADD	MPY	LD	ADD	103	105	6	12
7				ADD	MPY	LD	103	105	6	12
8					ADD	MPY	103	105	6	12
9						ADD	103	105	4E h	12

5 (a). Describe host port interface and explain its signals.

Sol:



Host port interface (HPI):

- Allows to interface to an 8bit or 16bit host devices or a host processor

Signals in HPI are:

- Host interrupt (HINT)
- HRDY
- HCNTL0 &HCNTL1
- HBIL
- HR/$\overline{w}$

Important signals in the HPI are as follows:

- The 16-bit data bus and the 18-bit address bus.
- The host interrupt, Hint, for the DSP to signal the host when it attention is required.
- HRDY, a DSP output indicating that the DSP is ready for transfer.
- HCNTL0 and HCNTL1, control signal that indicate the type of transfer to carry out. The transfer types are data, address, etc.
- HBIL. If this is low it indicates that the current byte is the first byte; if it is high, it indicates that it is second byte.
- HR/$\overline{w}$ indicates if the host is carrying out a read operation or a write Operation

5 (b). Briefly describe the following instructions of TMS320C54XX processors with an example.

i) MAC *AR5, +*AR6+, A, B ii) RPT Smem iii) RPTB iv) BANZ v) MAS *AR3-, *AR6+, B, A

Sol:

i) B ← A+ (*AR5) X (*AR6), After this operation, AR5 and AR6 are incremented

ii) instruction following the RPT instruction will be executed (Smem)+1 times

iii) a block of instructions are repeated (k+1) times , where k is the content of BRC

iv) Branch if auxiliary reg not equal to zero

v) A ← B+ (*AR3) X (*AR6) , this operation, AR3 is decremented and AR6 is incremented

6 (a). Draw the functional diagram of multiplier/ adder unit of TMS320C54XX processors and explain.

Sol:

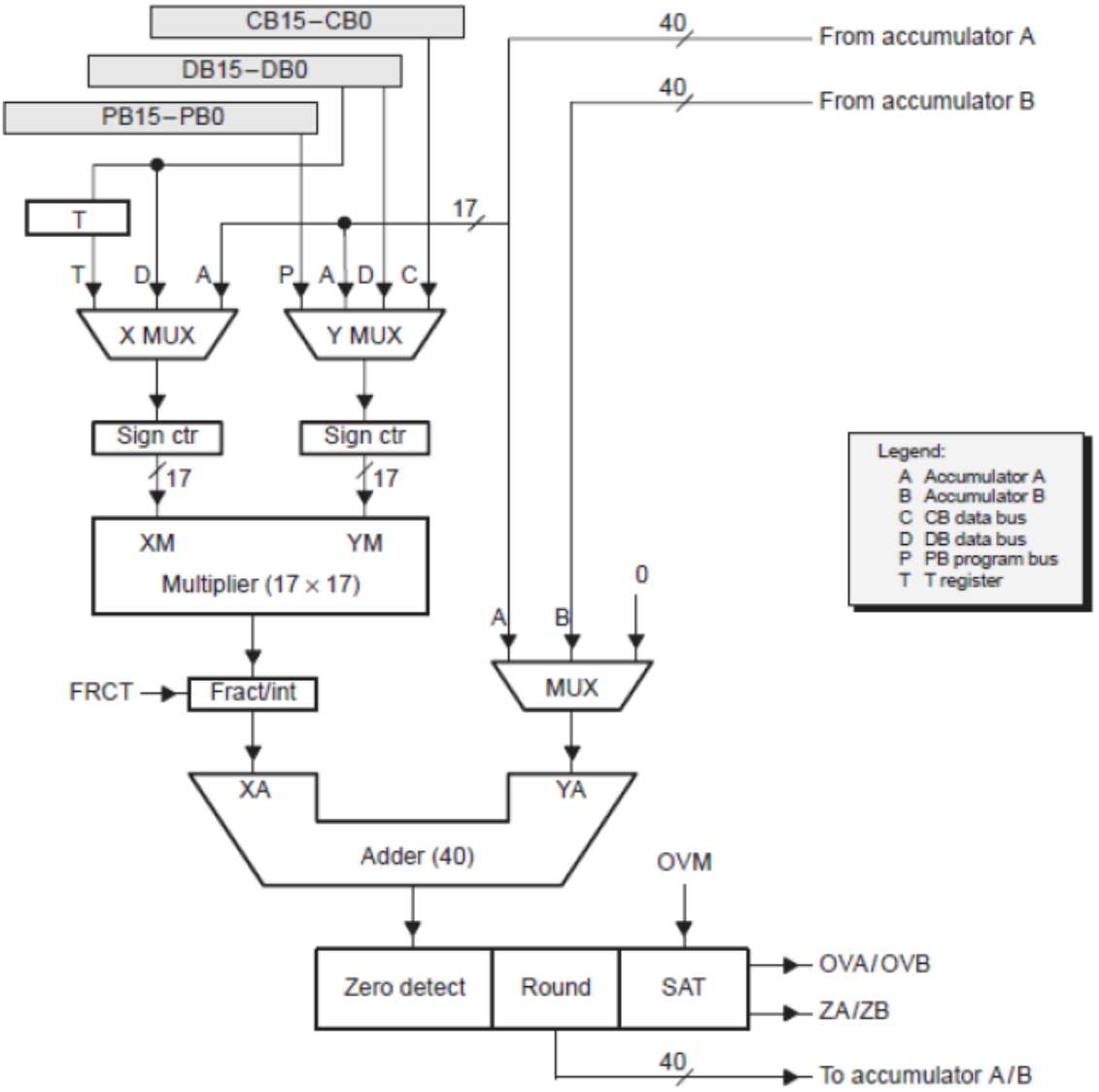


Figure 3.4. Functional diagram of the multiplier/adder unit of TMS320C54xx processors.

Multiplier/adder unit: The kernel of the DSP device architecture is multiplier/adder unit. The multiplier/adder unit of TMS320C54xx devices performs 17 x 17 2's complement multiplication with a 40-bit addition effectively in a single instruction cycle.

In addition to the multiplier and adder, the unit consists of control logic for integer and fractional computations and a 16-bit temporary storage register, T. Figure 3.4 show the functional diagram of the multiplier/adder unit of TMS320C54xx processors.

The compare, select, and store unit (CSSU) is a hardware unit specifically incorporated to accelerate the add/compare/select operation. This operation is essential to implement the *Viterbi* algorithm used in many signal-processing applications.

The exponent encoder unit supports the EXP instructions, which stores in the T register the number of leading redundant bits of the accumulator content. This information is useful while shifting the accumulator content for the purpose of scaling.

6 (b). Draw and explain the functional diagram of Barrel shifters of TMS320C54XX processors

Sol:

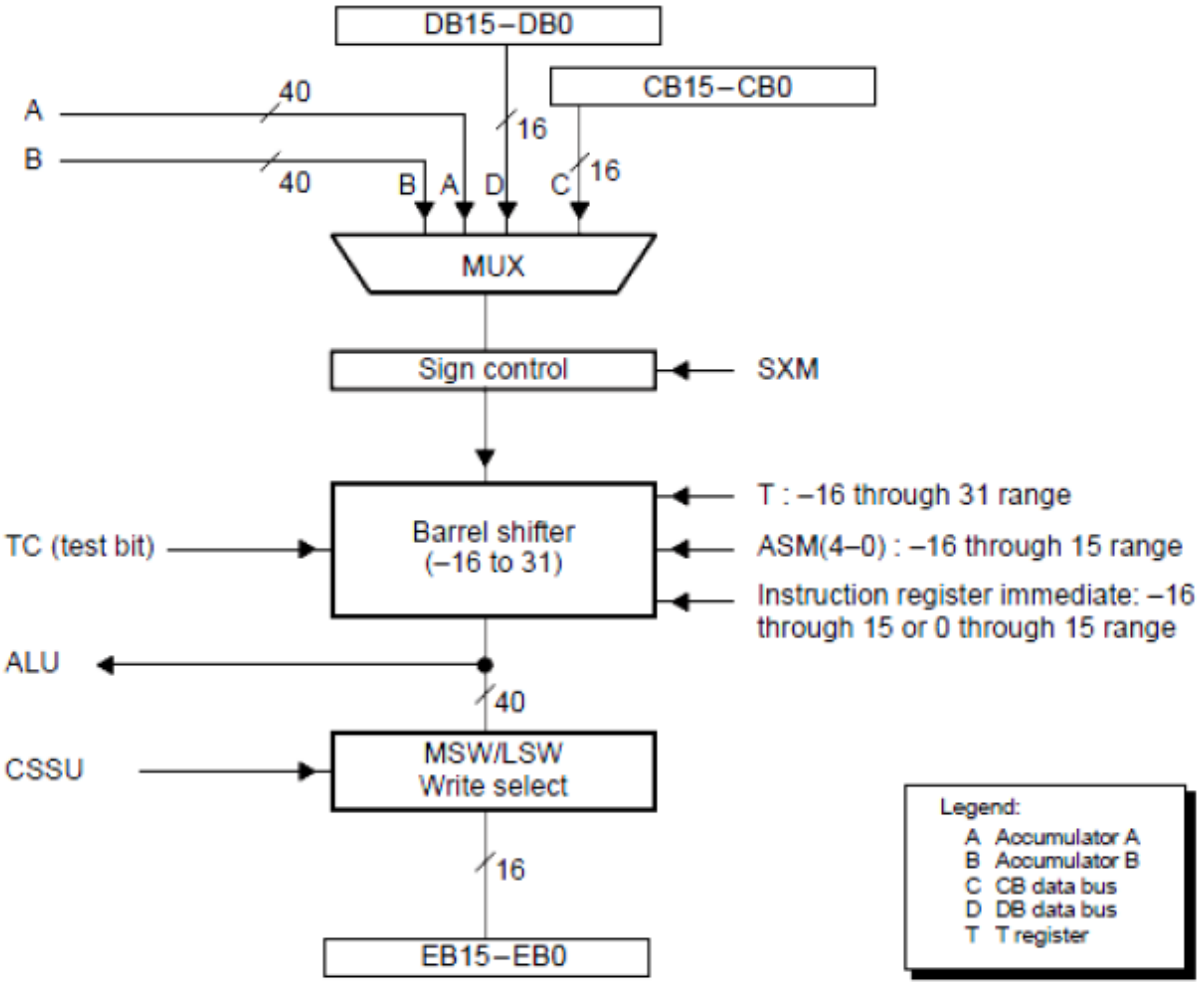


Figure 3.3. Functional diagram of the barrel shifter

Barrel shifter: provides the capability to scale the data during an operand read or write. No overhead is required to implement the shift needed for the scaling operations. The'54xx barrel shifter can produce a left shift of 0 to 31 bits or a right shift of 0 to 16 bits on the input data. The shift count field of status registers ST1, or in the temporary register T. Figure 3.3 shows the functional diagram of the barrel shifter of TMS320C54xx processors.

The barrel shifter and the exponent encoder normalize the values in an accumulator in a single cycle. The LSBs of the output are filled with0s, and the MSBs can be either zero filled or sign extended, depending on the state of the sign-extension

mode bit in the status register ST1. An additional shift capability enables the processor to perform numerical scaling, bit extraction, extended arithmetic, and overflow prevention operations.