A New Approach to Design of an Address Generation Unit in a DSP Processor

Kabiraj Sethi and Rutuparna Panda

Department of Electronics & Telecommunication Engg.,
VSS University of Technology, Burla – 768018. India
Fax: 91-0663-2430204 ; Email: r_ppanda@yahoo.co.in

***Abstract*—This paper presents the behavioral model of an Address Generation Unit (AGU) in a DSP Processor whose instructions are almost compatible with the Motorola DSP56002. The proposed AGU unit can handle 4 different types of arithmetic – linear addressing, modulo addressing, wrap around modulo addressing and reverse carry addressing. It also handles various means of calculating addresses as post/pre increment/decrement by a number. The novelty in this proposal is that it can address 2 different memories, where 2 new addresses are calculated concurrently. The central idea behind this design is address sequence generation by means of reverse carry addition, the use of modulo adder and offset adder. The designed AGU circuit generates the actual address as per the given set of inputs. Simulation results are compared with the theoretical data and found correct. The designed AGU may be implemented in a DSP Processor with optimized power and speed.**

Key words-VLSI design; Address Generation Unit; ALU; VHDL; DSP Processor.

I. INTRODUCTION

The present work aims at design and simulation of an Address Generation Unit (AGU) using VHDL [1-6]. From the literature [7-12], it is found that the main thrust has been to optimize speed, power and area of a DSP processor. The AGU is considered to be an important part of a DSP processor. Several researchers have investigated new design techniques for the design of AGU and related units. J Eyre and J Bier [13] have described utility of DSP processors in digital home audio, for down loading audio from internet and voice over internet. J Eyre [14] has presented a framework for understanding the DSP processor architecture including the interchange of architectural technique between DSP and general purpose processor. Eric Tell has carried out the Thesis work on the design of a domain specific DSP processor [15]. The first part of the Thesis gives different steps of design process and motivates the design decision made for the processor. The second part of the Thesis discusses design specifications. The evolution of DSP processors has been discussed by J Eyre and J Bier [16] and they have expected to see a continuing evolution of DSP processors as par with the change in technology. P John et al. have proposed a technique for generating an address for accessing a target location in a circular buffer of length L Bytes [17]. They [17] have provided means for generating an absolute address, wrapped address, a wrapped control signal and specify as the address of the target location.

E Quentin et al. [18] have investigated a reverse addition means for adding binary words in most significant to least significant bit order with the over flow or carry bit propagated to the left. This led to generation of a bit reverse address sequence that is mapped in to a closed apace. S Powel et al. [19] have presented three optimization techniques to reduce the energy in the data path, memory system and the instruction cache data bus. J Edwin et al. [20] explained the various DSP architectures and implementation. They also discussed the state-of-the art and examine the issues pertaining to performance of DSP processors. H Gustafsson [21] has described the design and implementation of AGU which can handle four different types of arithmetic. He [21] has also discussed the RTL model of the AGU that is one of the main parts of the DSP processor. All these techniques more or less emphasize either design of a single unit or design of complete DSP processor. This has motivated the authors to present a new approach for design of an AGU.

The Address Generation Unit (AGU) is one of three execution units on the DSP56002 core [22-24] as shown in the Fig.1. The AGU performs the effective address calculations (using integer arithmetic) necessary to address data operands in memory and contains the registers used to generate the addresses. To minimize address-generation overhead, the AGU operates in parallel with other chip resources. It implements four types of arithmetic: Linear, Modulo, Reverse-carry and Multiple wrap-around modulo.

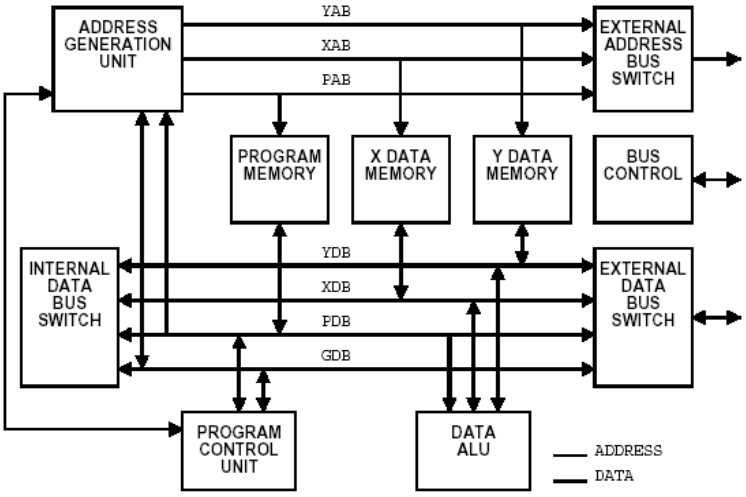

Figure 1. Block diagram of Motorola 56k DSP.

AGU Architecture

The AGU is divided into two identical halves (shown in Fig. 2), each of which has an address arithmetic logic unit (AALU) and four sets of three registers. They are the address registers (R0 - R3 and R4 - R7), offset registers (N0 - N3 and N4 - N7), and the modifier registers (M0 - M3 and M4 - M7). The eight Rn, Nn, and Mn registers are treated as register triplets, e.g., only N2 and M2 can be used to update R2. The eight triplets are R0:N0:M0, R1:N1:M1, R2:N2:M2, R3:N3:M3, R4:N4:M4, R5:N5:M5, R6:N6:M6, and R7:N7:M7 [23].

The two arithmetic units can generate two 16-bit addresses in every instruction cycle for two of the XAB and YAB. The two independent address ALUs work with the two data memories to feed the data ALU two operands in a single cycle. Each operand may be addressed by an Rn, Nn, and Mn triplet. The two Address ALUs are identical in the sense that each contain a 16-bit full adder (called offset adder), which can add 1) plus one, 2) minus one, 3) the contents of the respective offset register N, or 4) the two's complement of N to the contents of the selected address register.

A second full adder (called modulo adder) can add the summed result of the first full adder to a modulo value, M or minus M, where M is stored in the respective modifier register. A third full adder (called a reverse-carry adder) can add 1) plus one, 2) minus one, 3) the offset N (stored in the respective offset register), or 4) minus N to the selected address register with the carry propagating in the reverse direction i.e., from the most significant bit (MSB) to the least significant bit (LSB). The offset adder and the reverse-carry adder are in parallel and share common inputs. Test logic determines which of the three summed results of the full adders is output.

The output of the offset adder gives the result of linear arithmetic (e.g., Rn ± 1; Rn ± N) and is selected as the modulo arithmetic unit output for linear arithmetic addressing modifiers. The reverse-carry adder performs the required operation for reverse-carry arithmetic and its result is selected as the address ALU output for reverse-carry addressing modifiers.

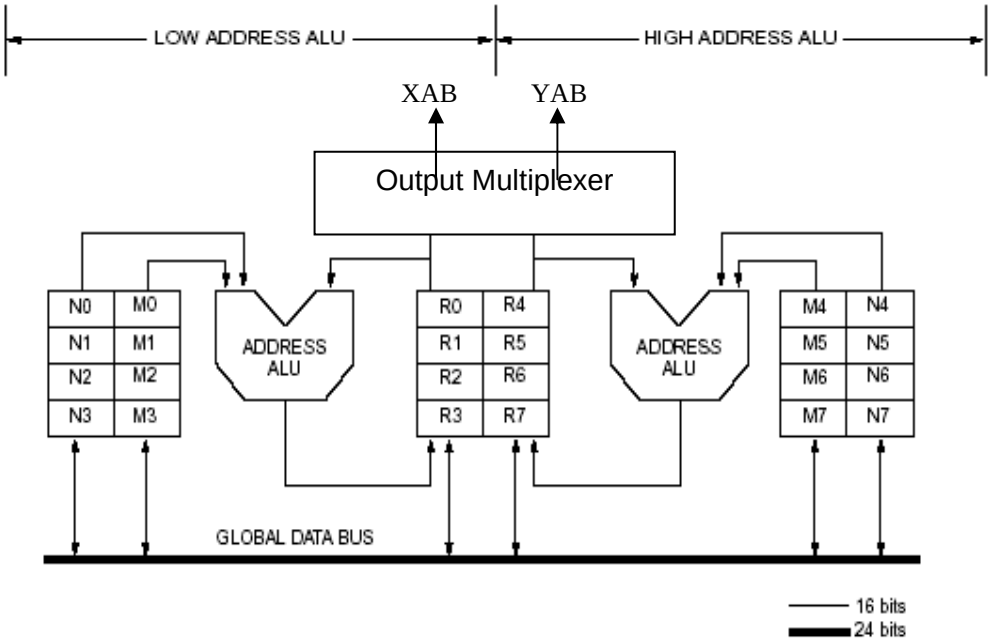

Figure 2. AGU Block Diagram.

Reverse-carry arithmetic is useful for 2^k -point fast Fourier transform (FFT) addressing. For modulo arithmetic, the modulo arithmetic unit will perform the function (Rn ± N) modulo M, where N can be one, minus one, or the contents of the offset register Nn. If the modulo operation requires wraparound for modulo arithmetic, the summed output of the modulo adder gives the correct updated address register value; if wraparound is not necessary, the output of the offset adder gives the correct result.

II. DESIGN APPROACH

This section describes the registers, addressing modes and address modifier types implemented in the design of AGU. Some ideas on the structure of the AGU were inspired from [21]. The programmer's view of the AGU is eight sets of three registers (shown in Fig. 3). These registers can act as temporary data registers and indirect memory pointers. The Mn registers can be programmed for linear addressing, modulo addressing, and bit-reverse addressing.

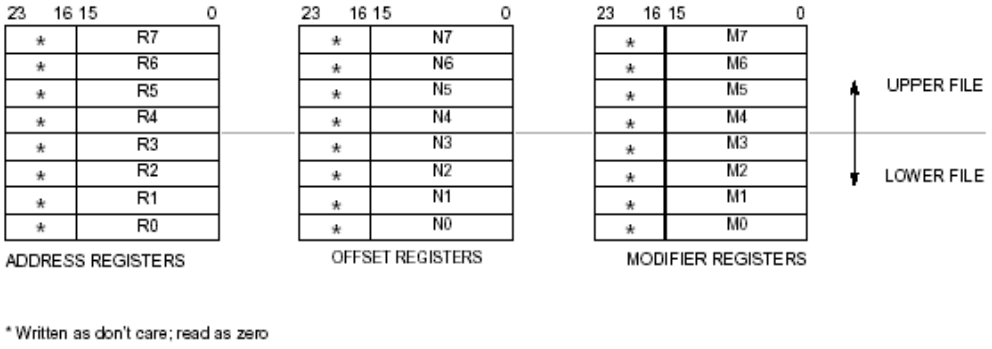

Figure 3. Registers of AGU.

Addressing Modes: The addressing modes that are implemented in this design are register direct and address register indirect. The address register indirect addressing modes use the registers in the AGU. When an address register is used to point to a memory location; the addressing mode is called "address register indirect". The addressing modes are summarized in Table 1.

TABLE 1. Addressing modes.

Addressing Modes	Uses Mn Modifiers	Assembler Syntax
Register Direct		
Data or Control Register	No	
Address Register Rn	No	
Address Modifier Register Mn	No	
Address Offset Register Nn	No	
Address Register Indirect		
No Update	No	(Rn)

Post increment by 1	Yes	(Rn)+
Post decrement by 1	Yes	(Rn)-
Post increment by Offset Nn	Yes	(Rn)+(Nn)
Post decrement by Offset Nn	Yes	(Rn)-(Nn)
Indexed by Offset Nn	Yes	(Rn)+(Nn)
Pre decrement by 1	Yes	-(Rn)

The instruction formats to implement the addressing modes are given in the following examples:

Addressing	Example
▪ No Update:	MOVE A1,X: (R0)
▪ Post increment By 1:	MOVE B0,Y: (R1)+
▪ Post decrement By 1:	MOVE Y0,Y: (R3)-
▪ Post increment By Offset Nn:	MOVE X1,X: (R2)+N2
▪ Post decrement By Offset Nn:	MOVE X:(R4)–N4,A0
▪ Indexed By Offset Nn:	MOVE Y1,X: (R6+N6)
▪ Pre decrement By 1:	MOVE X: –(R5),B1

A portion of the data bus movement field in the instruction specifies the memory space to be referenced. The contents of specific AGU registers that determine the effective address are modified by arithmetic operations performed in the AGU. The type of address arithmetic used is specified by the address modifier register, Mn. Examples for Addressing modes are found in [22, pp. 4-16]. Examples for Address Modifier Arithmetic Types are presented in [22, pp. 4-26].

Address Calculation:

Reverse Carry Modifier

If the +Nn addressing mode is used with the address modifier and Nn contains the value $2^{(k-1)}$ (a power of 2), addressing modifier is equivalent to bit reversing the k LSBs of Rn, incrementing Rn by 1, and bit reversing k LSBs of Rn again.

When M0 = 0 for reverse-carry addressing, modulo value M = 1.

Let N0 = 8 and R0 = 64

$Nn = 2^{(k-1)} = 2^3$ => k-1= 3 and k = 4bits

Now lower boundary =L × (2^k), Where $2^k \geq M$

Therefore, k=4 and lower address boundary must be multiple of 16. It can be chosen as 0, 16, 32, 64, etc. For this example, L is arbitrarily chosen to be 4, making the lower boundary 64.

Then the upper boundary = lower boundary + (2^k) -1

= 64+ 16- 1= 79

Steps: 1. Bit reverse k LSBs of R0.

R0 = 64 = 0100 0000 (64)

0000

2. Increment R0 by 1. + 1

0100 0001

3. Bit reversing again

k LSBs of R0 0100 1000 (72)

The procedure continues until we get the result within the specified boundary.

0100 1000 (72)

0100 0001

+ 1

0100 0010

0100 0100 (68)

0100 0010

+ 1

0100 0011

0100 1100 (76)

Modulo address modifier

For modulo address modifier, let M0 = 19
For +Nn addressing mode, let N0 = 5
For post increment by offset Nn addressing mode

$2k \geq M$

$\Rightarrow \quad k = 5.$

Lower boundary can be chosen as $L \times (2^k)$, and for L = 2; it is 64. Upper boundary = 64 + 20 – 1 = 83.

Let	R0 = 75 =	0100	1011
	M0 = 19 =	0001	0011
	N0 = 5 =	0000	0101
	M = 20 =	0001	0100

Then, (R0 + N0) =		0100	1011	(75)
	+	0000	0101	
		0101	0000	(80)

From the value of the Modifier register, the most significant 1 is found to be in the 4th bit position which is the output of the priority encoder. Responsive to this the selection circuit (SELECT1) picks out the carry bit c (4) =1 at the same bit position from the adder. This carry bit is provided to the inverter and one input of the OR gate. So, output of the inverter is 0.The selection circuit (SELECT2) picks out carry bit at same bit position c (4) from Adder/Subtractor and provide it to another input of OR gate. So, output of OR gate is 1.It indicates now the absolute address has crossed the upper boundary.

As N (15) = 0, the sign bit of the offset register. It indicates subtraction operation in the Adder/Subtractor. Hence, WRAP=1 and the output is the wrapped address instead of the absolute address.

(R0+ N0)		0101	0000	(80)
(M)	-	0001	0100	(20)
		0011	1100	(65)

The modifier value is decoded for M=20 and the decoder output is 0.Therefore, the Address ALU output is selected as the wrapped address.

III. DESIGN FLOW

The main objective of this part of the project is to provide a simple, fast and efficient hardware system for generating addresses for accessing circular buffers. A further intention is to provide address generation apparatus which allows successively accessed buffer locations to be separated by an arbitrary number of address locations.

Xilinx Project Navigator tool (ver. ISE7.1i) [25] is used to write the VHDL codes and synthesize it. ModelSim XE-III 6.0d is used for testing the design.

AGU Block Structure: The AGU have ten control signals, one global data bus and two address buses and two data buses. Block Structure over the designed model is displayed in Fig. 4. The control signals can be divided into several blocks. All signals that contain lower only affect the lower file, i.e. register triplets 0 to 3. In the same way all signals that contain upper only affects the upper file.

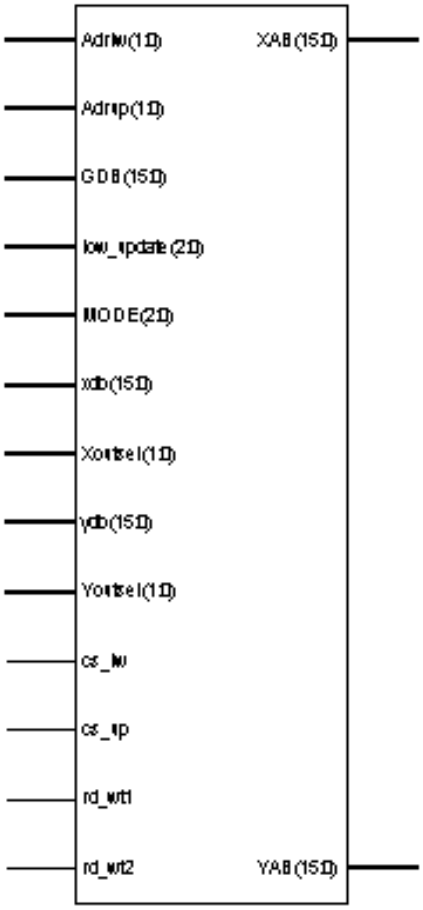

Figure 4. Block Structure of AGU (Designed).

Address low (upper) is a 2-bit vector that decides which register files low (upper) to be selected. Two output selection signals decide whether output will be XAB or YAB. Two chip select signals are chosen from the upper AALU and lower AALU. Two read/write control signals select the location for read and write operation. Fig. 5 shows the schematic diagram of AGU.

Blocks of AGU:

AGU contains the following parts (shown in Fig.5):

Address Register Files (R0-R3 and R4-R7): The eight 16-bit address registers, R0 - R7, can contain addresses or general-purpose data. The 16-bit address in a selected address register is used in the calculation of the effective address of an operand. When supporting parallel X and Y data memory moves, the address registers must be thought of as two separate files, R0 - R3 and R4 - R7. Rn can be pre-updated or post-updated according to the addressing mode selected. If an Rn is updated, modifier registers, Mn, are always used to specify the type of update arithmetic. The form of address register modification performed by the modulo arithmetic unit is controlled by the contents of the offset and modifier registers.

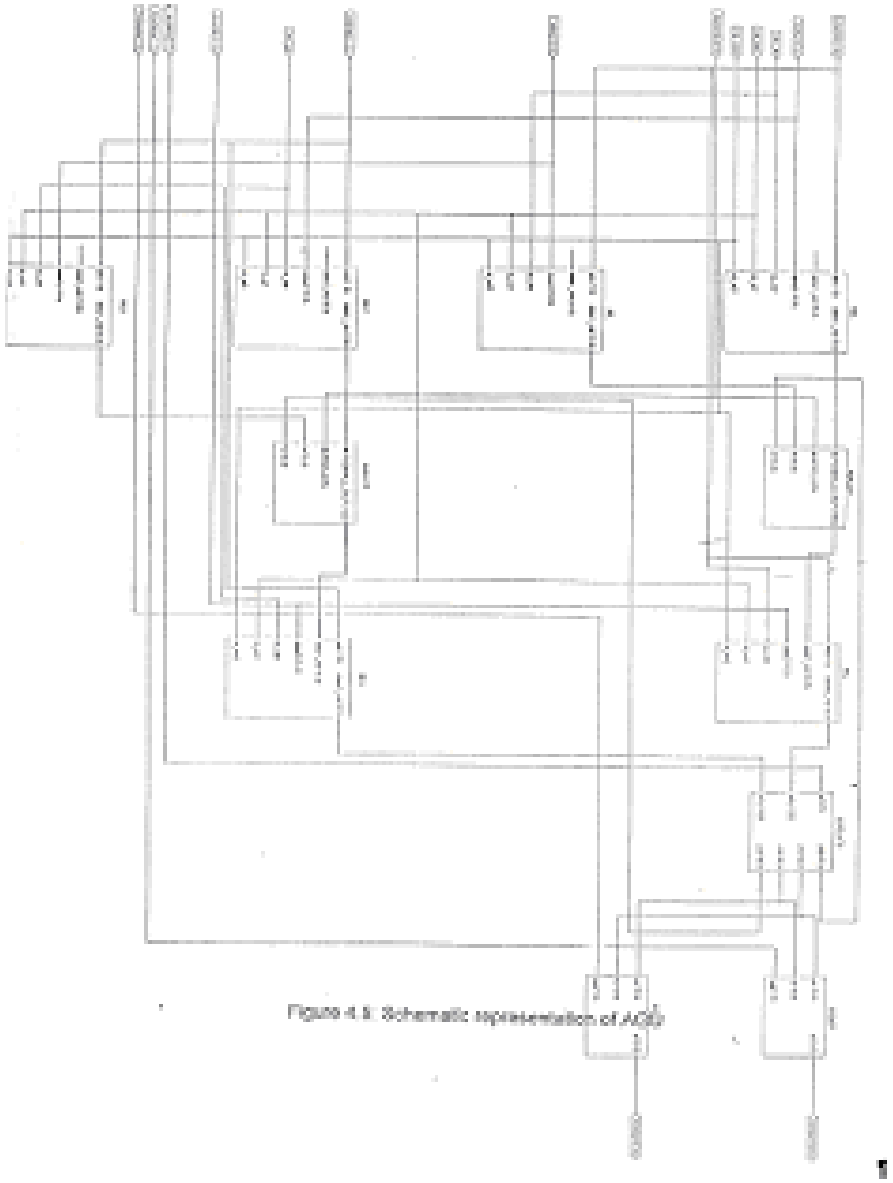

Figure 5. Schematic diagram of Address Generator Unit.

Offset Register files (N0-N3 and N4-N7): The eight 16-bit offset registers, N0-N7, can contain offset values used to increment/decrement address registers in address register update calculations or can be used for 16-bit general-purpose storage.

Modifier Register Files (M0-M3 and M4-M7): The eight 16-bit modifier registers, M0 - M7, define the type of address arithmetic to be performed for addressing mode calculations, or they can be used for general-purpose storage. For modulo arithmetic, the contents of Mn also specify the modulus. Fig.6 shows the Register File block structure used in the above six Register Files.

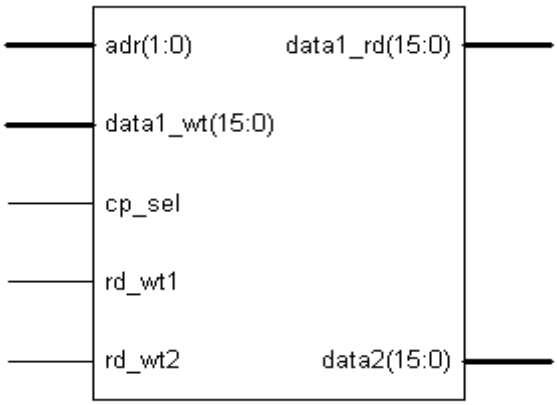

Figure 6. Register File.

Output multiplexers: The two address output multiplexers select the address for the XAB, YAB, where the address originates from the R0 - R3 or R4 - R7 registers. Fig. 7 represents the block structure of Output multiplexers.

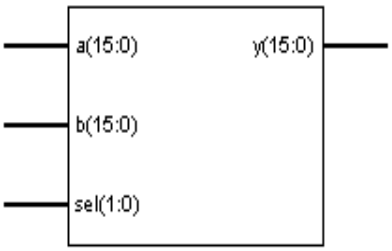

Figure 7. Output multiplexer.

Logic (Keep/Update): It decides whether the lower (upper) file shall be updated or kept as unmodified during the current operation. Fig. 8 represents the block structure of Output multiplexer. Keep/Update Logic.

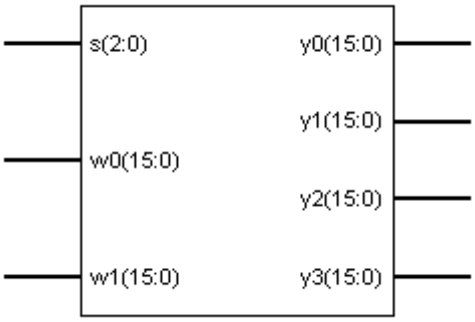

Figure 8. Keep/Update Logic.

Address Arithmetic Logic Unit (AALU): Each Address ALU comprises an offset adder for generating an absolute address, a modulo adder for generating a wrapped address which maps the absolute address into the address space within the boundaries of the buffer, a reverse carry adder for generating a bit-reversed address and/or an address sequence that is mapped into a “closed” space [17], and some internal logic.

In AGU, the contents of the three registers represent, respectively, the length of the buffer, the current address designated by the pointer, and the offset from the current buffer address to the target address. In AALU, the adder generates the absolute address by adding the current address of the pointer to the offset. The adder/subtractor for modulo adder generates the wrapped address by either adding to the absolute address or subtracting from the absolute address of the length of the buffer, depending on the direction of the offset. Logic operating on the carry bits from the adder and the adder/subtractor detects the generation of an absolute address outside the boundaries of the circular buffer and provides as the target address of the wrapped address instead of the absolute address.

Test logic is implemented for selecting as the target address, the absolute address, wrapped address or bit-reverse address.

Fig.9 represents structure of AALU. Fig.10 and Fig.11 show the block structure and schematic view of AALU, respectively.

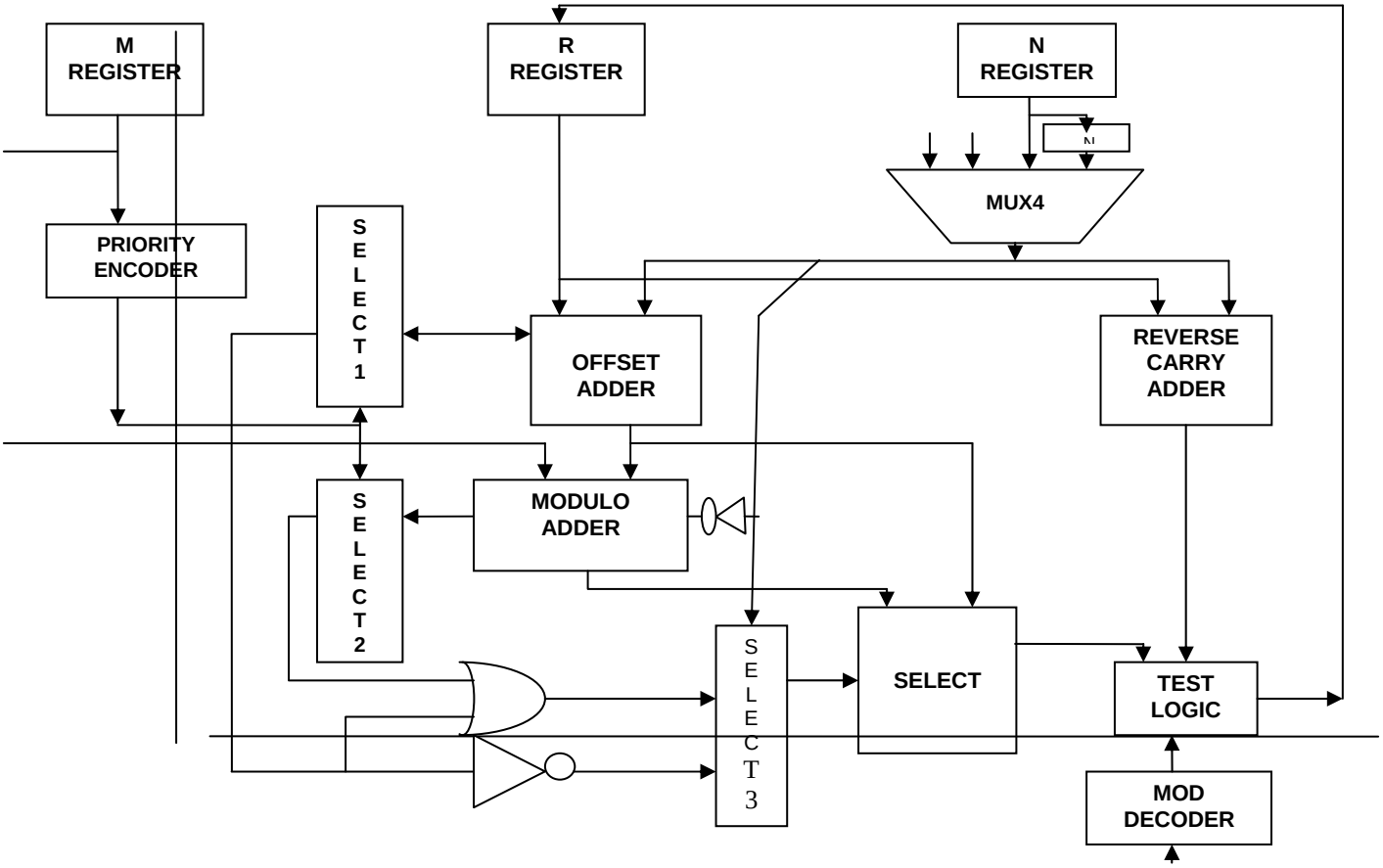

Figure 9. Structural representation Address ALU.

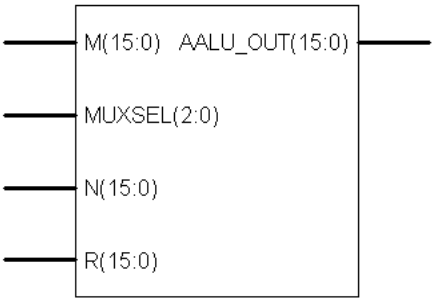

Figure 10. Address ALU.

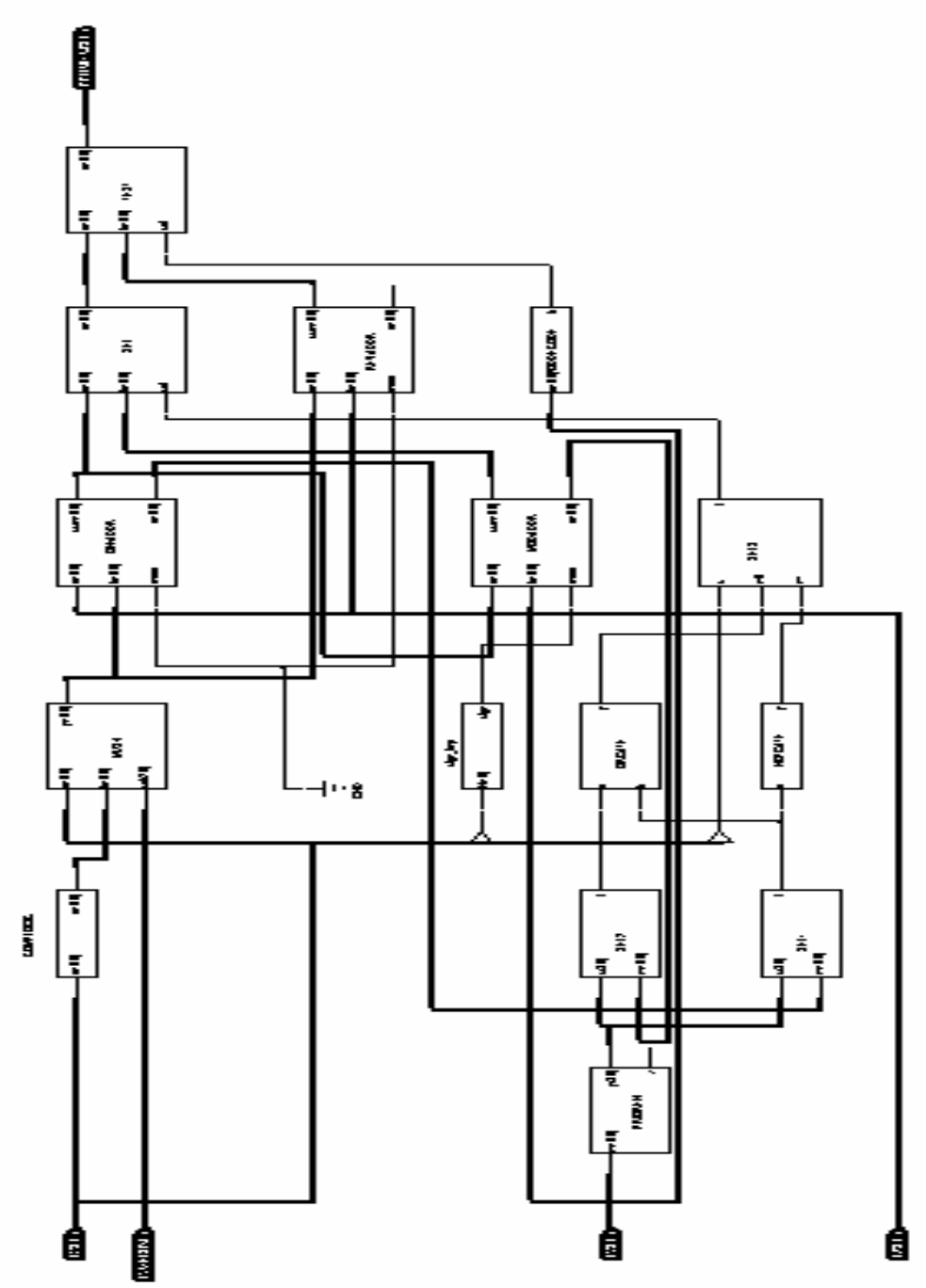

Figure 11. Schematic Diagram of AALU.

Blocks of Address Arithmetic Logic Unit(AALU)

OFFSET ADDER is implemented with ripple-carry logic. It therefore generates as its output a set of sum bits and a set of carry bits. The sum is provided to an adder/subtractor and to a selection circuit.

REVERSE CARRY ADDER includes a reverse addition means for adding binary sequence. By propagating the carry bit in opposite direction than that is used in normal addition and by adding binary words from most significant digit to least significant digit order, a bit reversed sequence is produced [18].

MODULO ADDER generates either the sum of its input or their difference [9,26], depending upon the state of the sign bit from the N register, to provide the modified or “wrapped” address. Fig 12 represents the block structure of. Offset/Reverse Carry Adder and Module Adder.

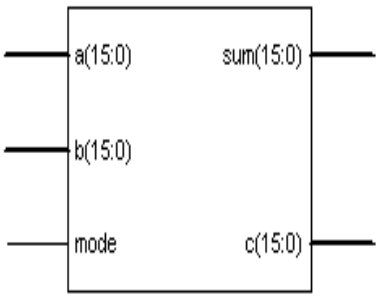

Figure 12. Adder circuit.

SELECTION CIRCUIT (SELECT) supplies the next buffer address to the address register R. Selection circuit is controlled by a WRAP signal. When the WRAP signal is a 1, the output is the WRAP address generated by adder/subtractor; and when WRAP signal is a 0, the output is the absolute address generated by the adder.Fig.13 shows the Select Circuit

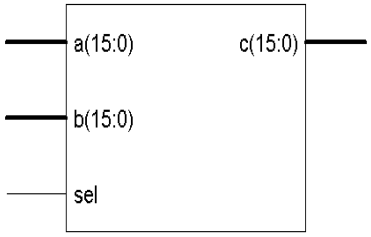

Figure 13. Select Circuit.

CONTROL CIRCUIT circuitry for generating the WRAP signal comprises of a pair of selection circuits (SELECT1 and SELECT2), a priority encoder, an inverter, an OR-gate and another selection circuit (SELECT3). Fig.14 represents the Control Circuit

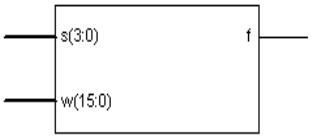

Figure 14: Control Circuit (SELECT1/2).

MODIFIER DECODER (MODDECODE): Modifier value is decoded in AALU and affects the operation. The output is fed to the test logic that determines which of the three summed output is the output to its associated address register file.Block structure of Modifier decoder logic is shown in Fig.15

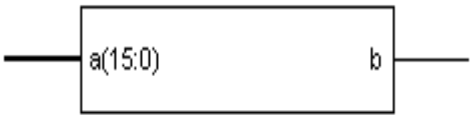

Figure 15. Modifier decoder logic.

TEST LOGIC consists of a multiplexer that selects from the summed output which is to be fed to the address register file for further update according to the specified addressing mode .Test logic block is given in Fig.16.

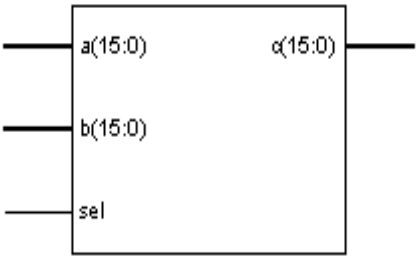

Figure 16. Test Logic.

IV. DESIGN DESCRIPTION AND OPERATION

GDB is the Global Data Bus and is 16 bits long. Two data buses, XDB and YDB along with GDB are used to assign different values to the register files in this design instead of using a single GDB and a data bus switch. To calculate the updated addresses, there are seven addressing modes that can be used. All the post update and the

no update modes will use the contents of the current address register as output while pre update mode will calculate the updated address first and uses the result as output. The register triplets are also split in two blocks, a lower file and an upper file. The benefits of this split-up are that two addresses can be calculated at the same time, provided that one address is in the upper file and the other is in the lower file. All 24 AGU registers are 16-bits wide. Each address register has its own offset register and modifier register, creating 8 register triplets. Every register can only work with the other registers in the same triplet. The address register gives the actual address, the offset register gives the offset from the base address and the modifier register specifies which modifier mode that shall be used.

The DSP has 4 different modifier modes, which specifies what kind of address arithmetic that shall be used. The AGU is also controlled with an addressing mode selector. The addressing mode specifies how the offset values shall be used.XAB is the X-memory Address Bus and it is an output signal, which is 16 bits long. YAB is the Y-memory Address Bus and it is an output signal, which is 16 bits long. The priority encoder in the AALU receives as its input the contents of Modifier register (M) and supplies, at its output, a signal which identifies the bit position of the most significant 1 in the contents of the M register. In response to that output signal from priority encoder, the SELECT1 picks out (from the carry bits of adder) the carry output bit which is from the same bit positions as the most significant 1in the M register. The selected carry bit is provided to the inverter and to one input of the OR-gate. The output of the inverter provide a signal indicating whether the new w absolute address has crossed the lower boundary of the buffer's memory area, when the offset is negative.SELECT2 operates in the same way as SELECT1 and is also under the control of the output of priority encoder. OR-gate provides the logical OR of the selected carry bits from the adder and the adder/subtractor. The output of the OR-gate indicates whether the new absolute address has crossed the upper boundary of the buffer, when the offset is positive.SELECT3 chooses the output of inverter or OR-gate responsive to the sign bit of the N register, to produce the WRAP signal.If the next address is intended to be "higher" than the present address, the sign bit in register N is 0, representing a positive offset. If the next address is intended to be "lower" than the present address, the sign bit is 1, representing a negative offset. The sign bit of register N is provided to the SELECT3.When the WRAP signal is a 1, the output is the WRAP address generated by modulo adder, and when WRAP is a 0, the output is the ABSOLUTE address generated by the offset adder.Modifier value is decoded by modifier decoder and fed to the test logic as a control signal that determines which of the three summed output is the output to the associated address register file.

Simulation results are produced in Figures 17-21.

SIMULATION AND RESULTS

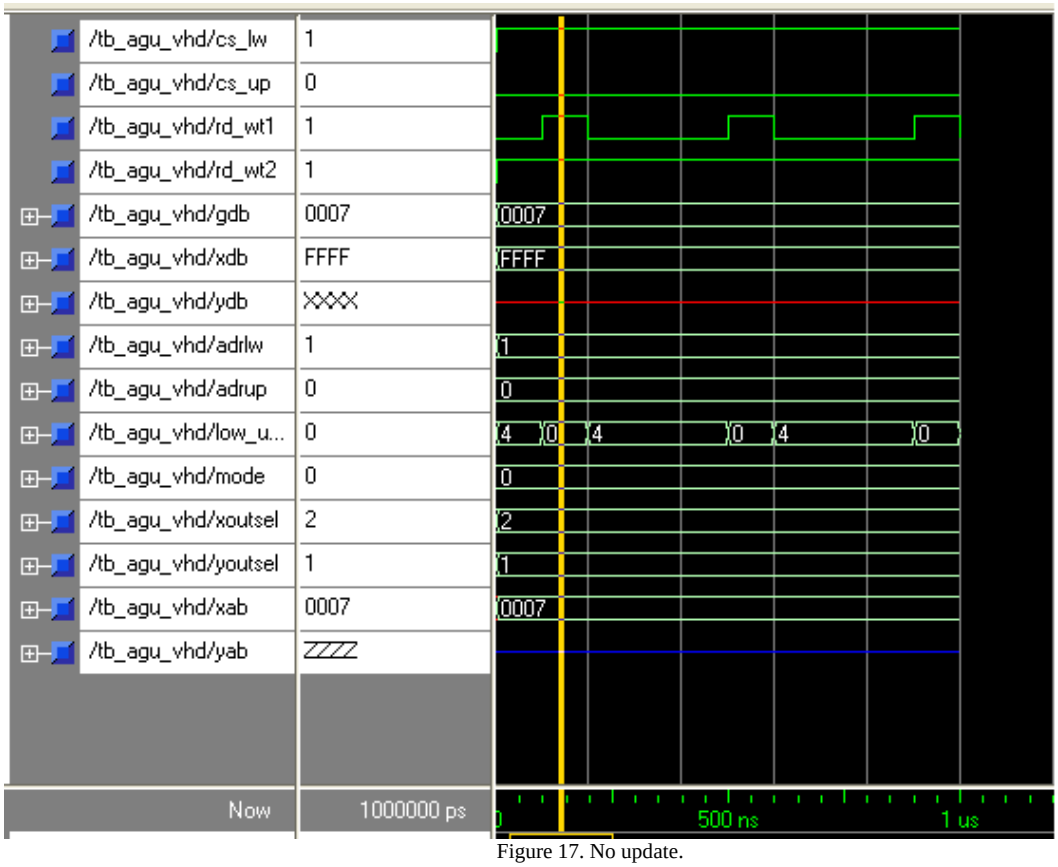

Figure 17. No update.

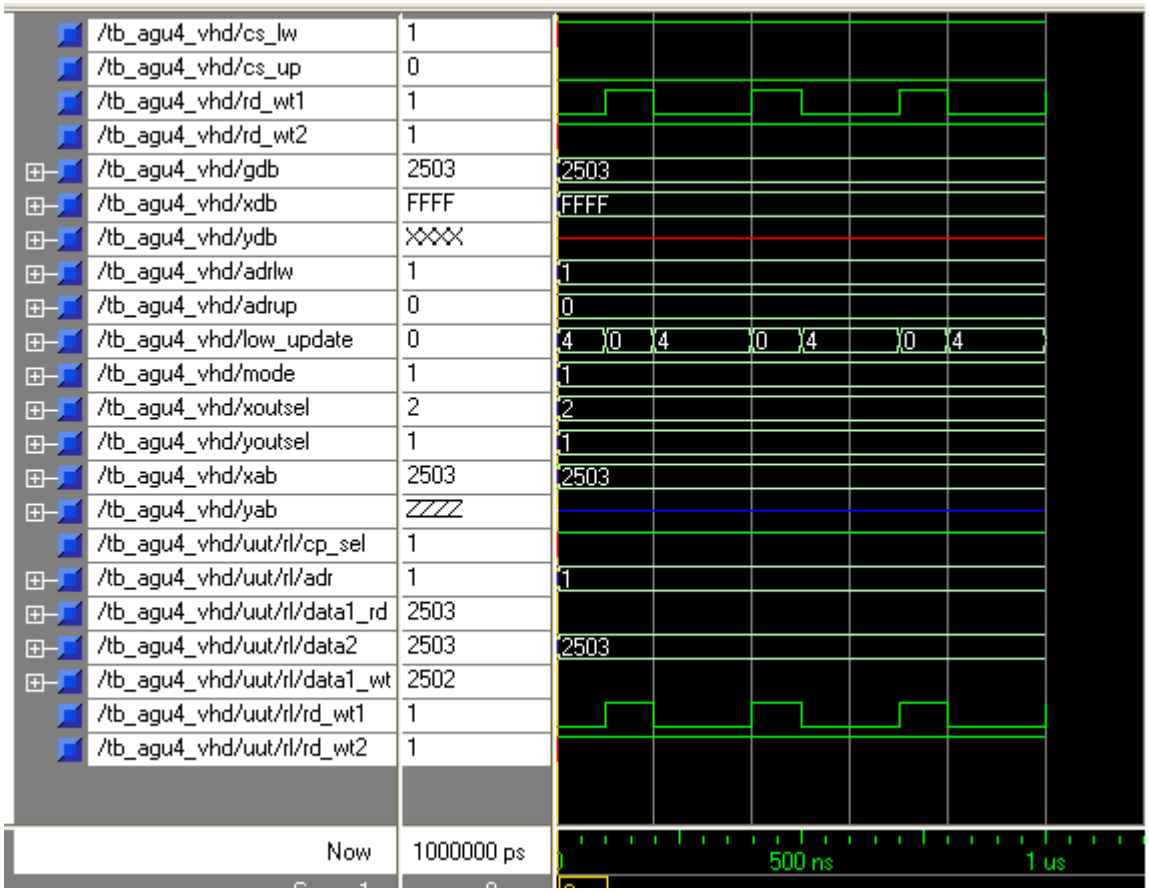

Figure 18. Post Decrement by 1.

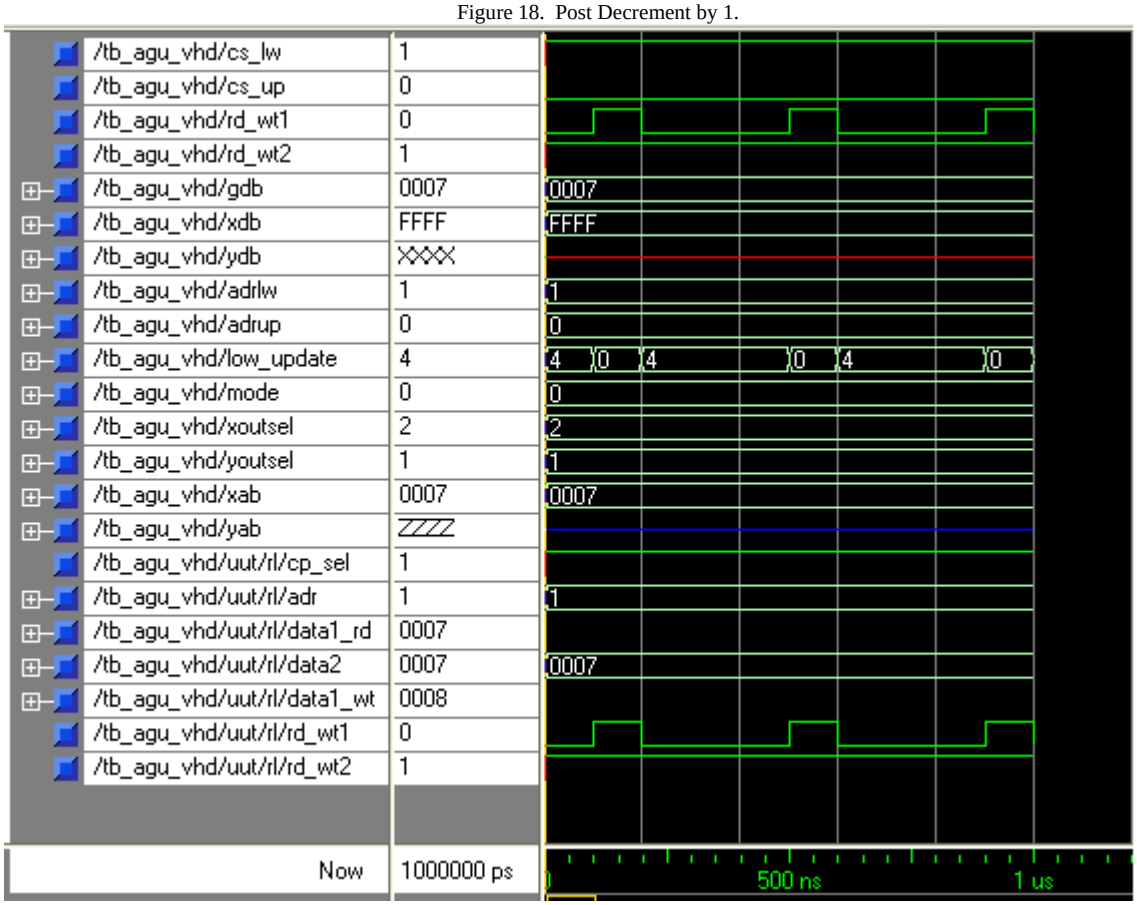

Figure 19. Post increment by 1.

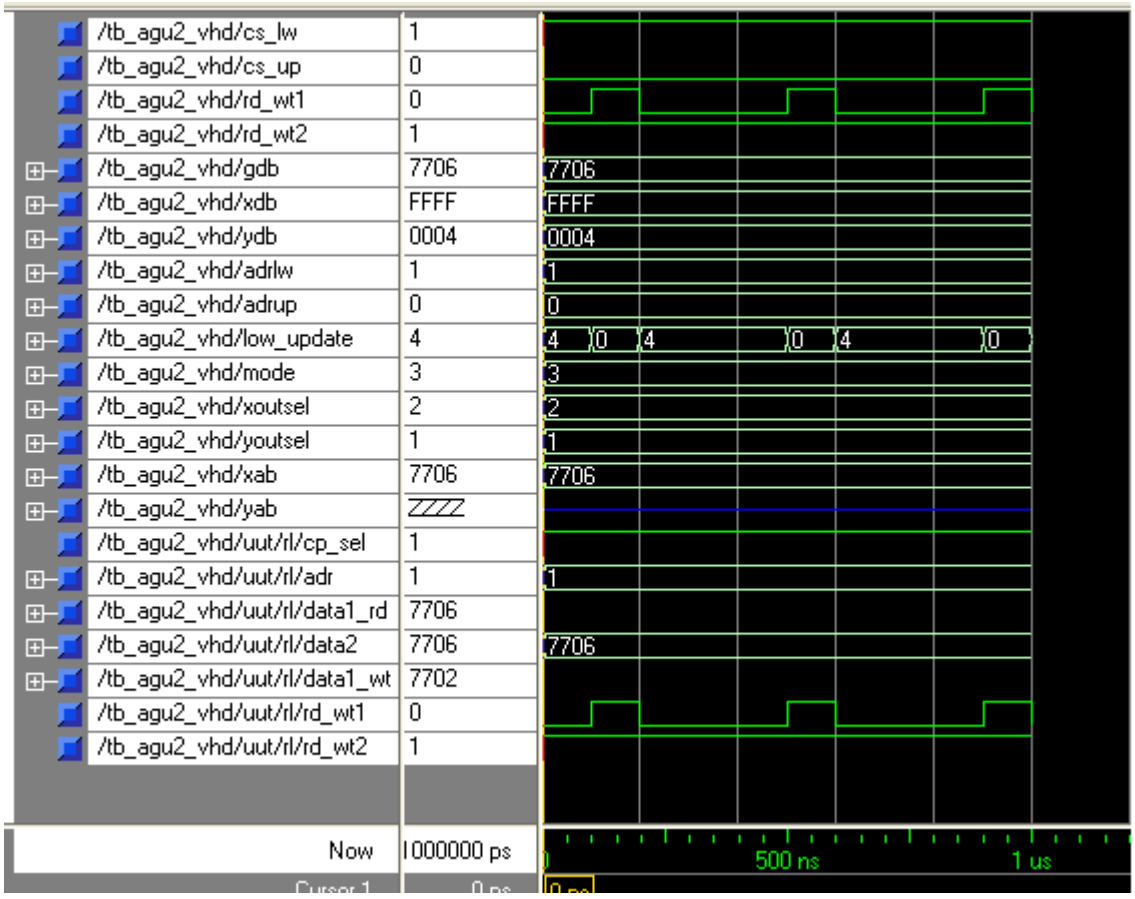

Figure 20. Post decrement by offset Nn.

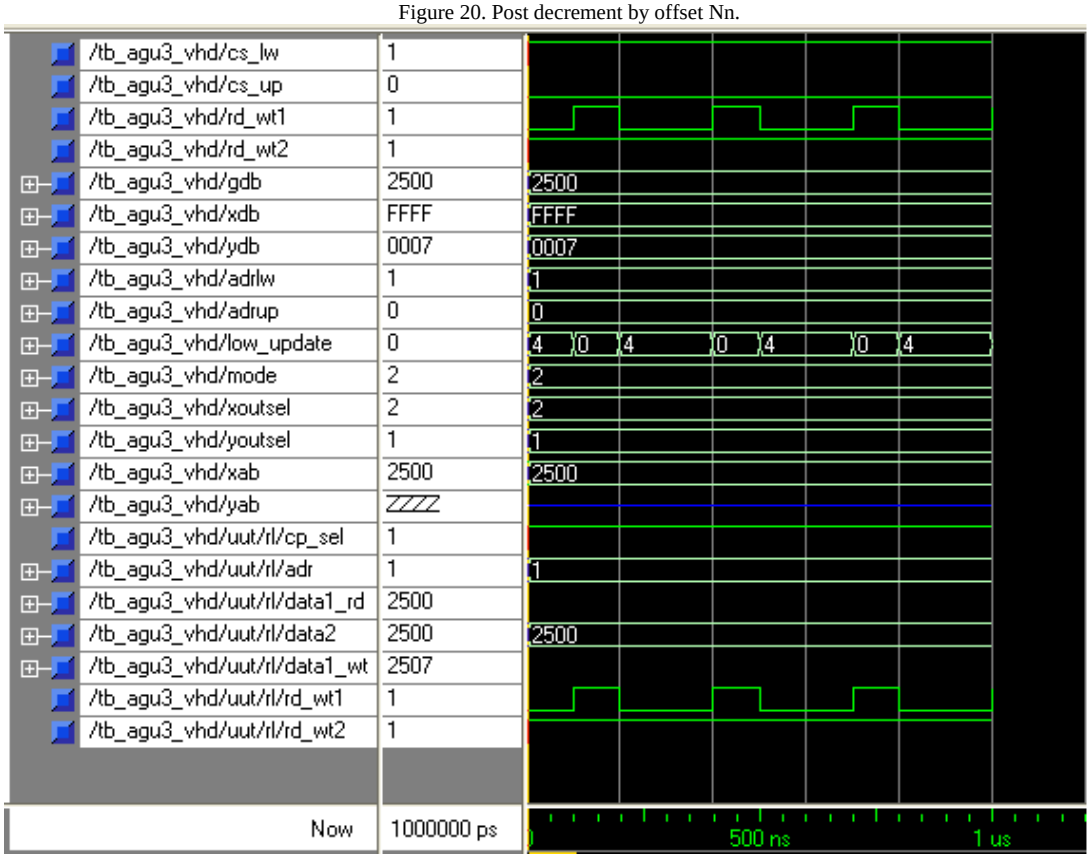

Figure 21. Post increment by offset Nn.

V. RESULTS VERIFICATION

LINEAR MODIFIER

(Post-increment by 1)

Muxsel	M	R	N	Output
0	FFFF	2978	0000	2979
		1000		1001
		6790		6791
		2543		2544
		2601		2602
		4735		4736
(Post-decrement by 1)				
1	FFFF	2678	0000	2677
		4735		4734
		4656		4655
		9631		9630
(Post-increment by offset N)				
2	FFFF	7706	0005	770B
		4735	0008	473D
		2500	0002	2502
		3200	0004	3204
		6000	0007	6007
		3002	0006	3008
(Post-decrement by offset N)				
3	FFFF	5643	0002	5641
		8768	0002	8766
		5678	0006	5672
		3002	0004	2FFE
		2678	0003	2675
(Indexed by offset N)				
2	FFFF	4367	0004	436B
		3200	0002	3202
		4578	0003	457B
		3577	0008	357F

MODULO MODIFIER

(Post-increment by offset N)

Muxsel	M	R	N	Output
2	21	75	15	69
	19	75	15	71
	19	80	15	76
	15	80	05	85
	15	65	05	70

REVERSE-CARRY MODIFIER

(Post-increment by offset N)

Muxsel	M	R	N	Output
2	0	64	08	72
		72	08	68
		68	08	76
		76	08	66

Performance evaluation is done and results are produced in Table 2.

TABLE 2: Performance evaluation.

Total equivalent gate count for design	17636
Maximum combinational path delay	11.262ns
Minimum period: 34.662ns (Maximum Frequency)	28.850MHz
Maximum Power Consumption	7mW at 3.3V V_{CC}

VI. CONCLUSION

The Motorola DSP56002 has been used as the architectural model for the implementation for reducing the risk of ending up with an erroneous design. The features, architectures and the instruction set are studied frequently and extensively throughout this paper. Top down approach is found logical for such a design and is adopted in this work. The design methodologies available in the literature are explored and the best possible solution is adopted for the proposed design. The design is optimized in terms of speed and low power consumption. The expected simulation results are obtained which are presented in Figs.17-21. The designed AGU circuit generates the actual address as per the given set of inputs and is discussed in result verification part of this paper. Table 2 displays the simulated data for the proposed AGU in terms of power and speed. Finally, we conclude that the proposed approach for design of AGU may be useful for design of DSP Processor.

REFERENCES

[1] Charles H Roth, Jr, Digital systems design using VHD, Brooks/ Cole Thomson learning, 2002.
[2] Douglas Perry, VHDL 3rd Edition, McGraw-Hill International Educations, 2008.
[3] Jayaram Bhaskar, A VHDL Primer, 3rd Edition, Pearson Education Asia, 2009.
[4] Peter J. Ashenden,The VHDL Cookbook, Dept. Computer Science University of Adelaide South Australia, July 1990..
[5] Volnei A.Pedroni, Circuit Design with VHDL, PHI publication, 2004.ISBN 81-203-2683-0.
[6] Peter J. Ashenden, The Designer's Guide To VHDL,Systems on Silicon, Morgan Kaufmann Publishers, an Imprint of Elsevier, 2002, ISBN : 1-55860-674-2.
[7] Jon G. Proakis, Dimitris G Manolakis, Digital Signal Processing, PHI Publication, 2008.
[8] J. L. Hennessy and D. A. Patterson, Computer Architecture a Quantitative Approach, Morgan Kaufman, 1996.
[9] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, A Design Perspec-tive, Prentice Hall, Upper Saddle River,NJ, 2003.
[10] M. Moris Mano, Computer Architecture, PHI publication, 2002.
[11] Peter Pirsch, Architecture for Digital signal processing, Wiley-Interscience Publication, 2005.
[12] Massoud Pedram, Design Technologies for Low Power VLSI in Encyclopedia of Computer Science and Technology, 1995.
[13] J. Eyre and J. Bier, "DSPs court the consumer," IEEE Spectrum Magazine, vol.36, pp.47-53, 1999.
[14] J. Eyre, "Digital signal processor derby," IEEE Spectrum Magazine, vol. 38, pp.62-68, June 2001.
[15] Eric Tell, A Domain Specific DSP Processor, LiTH-ISY-EX-3209, Linköping 2001.
[16] J. Eyre and J. Bier, "The Evolution of DSP Processors, Berkeley Design Technology," Inc. (BDTI). IEEE Signal Processing Magazine, vol.17, pp.43-51, March 2000.
[17] John P, Roesgen, Easton and Mass, "MODULO ADDRESS GENERATOR," United State Patent, Patent Number - 4,800,524. Jan 1989.
[18] Quentin E, Dolecek, Burtonsville,Md. , "ADDRESS SEQUENCE GENERATION BY MEANS OF REVERSE CARRY ADDITION," United State Patent, Patent Number - 4,974,188. Nov 1990.
[19] S. Powell et al, "Estimating power dissipation of VLSI signal processing chips- the PFA technique," VLSI signal processing IV, pp.250-259, 1990.
[20] Edwin J. Tan, Wendi B. Heinzelman, DSP Architectures: Past, Present and Future, Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, 2004.
[21] Henrik Gustafsson, Behavioral model of an address generation unit, LiTH-ISY-EX-ET-0271-2003 Linköping 2003.
[22] Motorola, DSP56000 24-Bit Digital Signal Processor Family Manual, DSP56KFAMUM/AD 1994.
[23] Motorola, DSP56300 Digital Signal Processor Family Manual, Chapter-4, Address Generation Unit, 1995.
[24] Motorola, DSP56002 24-Bit Digital Signal Processor Family Manual, DSP56KFAMUM/AD 1994.
[25] www.xilinx.com
[26] Neil H. E. Weste, David Harris, Ayan banerjee, CMOS VLSI DESIGN, A Circuits and Systems Perspective, Pearson Education, 2006.

AUTHORS PROFILE

Kabiraj Sethi is presently a faculty in the Department of Electronics and Telecommunication Engineering, VSS University of Technology Burla. His area of research interests includes –VLSI signal processing and Computer architecture .

Dr. Rutuparna Panda was born in 1963. He received B.Sc Engg. and M. Sc. Engg. degrees from UCE Burla in 1985 and 1988, respectively. He obtained Ph.D.(Engg) from IIT,KGP in 1998. He is currently a Professor in the Department of Electronics and Telecommunication Engineering and HOD, Computer Science and Engineering/ Information Technology, VSS University of Technology Burla. He has guided 21 M.Tech Theses and 4 Ph.D. Theses. He has over 70 papers in International/National Journals and conferences. His area of research interests includes – Bioinformatics, Biomedical Image Fusion, Digital signal/image processing, VLSI signal processing.