CG3207 Computer Architecture

Semester 1 2014/2015

Lab 2 Group Report

Group No: **19**	Name	Matriculation Number
Group Member #1	Wong Zhi Qian Sebastian	A0101856J
Group Member #2	Chew Yi Xiu	A0101742W
Group Member #3	Angela Heng	A0101695H

1: Number of Cycles for each operation

Operation	Cycles
ADD	1
SUB	1
SLT	1
SLTU	1
MULT	33
MULTU	33
DIV	33
DIVU	33
SLL	1
SRL	1
SRA	1
AND	1
OR	1
XOR	1
NOR	1

2: Special Algorithm / Optimization

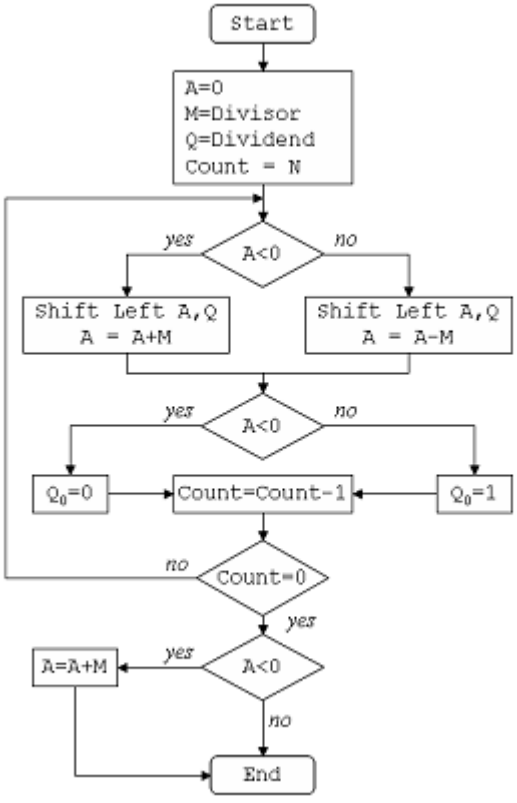
Multiply - MULT/MULTU

Signed multiplication (MULT) is done by checking the sign bit of both operands and determining the sign of the product.

If an operand is negative, it will be converted into unsigned binary before unsigned multiplication is performed. Upon obtaining the result of the unsigned multiplication, if the result of the multiplication is supposed to be negative (For instance, negative number x positive number), the product will be converted back into 2's complement form.

Unsigned multiplication (MULTU) is done by multiplying the 2 operands directly and obtaining the product.

Division - DIV / DIVU



Picture 1: Non-restoring Divsion Algorithm

Action	A	Q	Count
Initial	000 000	101 110	6
A > 0 => SHL (AQ)	000 001	011 10□	
A = A-M	101 010	011 10□	
A < 0 => Q0 = 0	101 010	011 100	5
A < 0 => SHL (AQ)	010 100	111 00□	
A = A+M	101 011	111 00□	
A < 0 => Q0 = 0	101 011	111 000	4
A < 0 => SHL (AQ)	010 111	110 00□	
A = A+M	101 110	110 00□	
A < 0 => Q0 = 0	101 110	110 000	3
A < 0 => SHL (AQ)	011 101	100 00□	
A = A+M	110 100	100 00□	
A < 0 => Q0 = 0	110 100	100 000	2
A < 0 => SHL (AQ)	101 001	000 00□	
A = A+M	000 000	000 00□	
A < 0 => Q0 = 1	000 000	000 001	1
A > 0 => SHL (AQ)	000 000	000 01□	
A = A+M	101 001	000 01□	
A < 0 => Q0 = 1	101 001	000 010	0
Count has reached Zero, So final steps			
A < 0 => A = A+M	**000 000**	**000 010**	
	Reminder	**Quotient**	

Picture 2: Example of how non-restoring division algorithm works

A check is being done on the two operands namely the dividend and the divisor. If the divisor is x”00000000”, it means that the results are undefined. Thus both result1, the quotient and result2, the remainder are set to x”XXXXXXXX”. If dividend is x”00000000” and divisor is any number other than x”00000000” then both the quotient and remainder will be x”00000000”. If dividend is a smaller than divisor, the quotient will be x”00000000” and remainder will be equal to the dividend.

Non-restoring division algorithm is used when dividend is a larger number than divisor. The diagram above, picture1 shows how non-restoring division works. At the start, a variable of a 32-bit number will be introduced and set to 0. This algorithm involves checking if A is smaller or

larger than 0 for two times every cycle. After each cycle is done, count will be incremented. Count is initially set to 0, for 32-bit registers used for dividend and divisor, a total number of 32 cycles is needed for this algorithm. In the first check, a left shift in both A and Q, and a 2's complement addition is required. In the second check, the last bit of Q is set to either 0 or 1. Then count increases. When count is 32, a last comparison is done to check if A is less than 0. If it is not less than 0, the final value of A will be a addition of A and M. The final result of A represents the value of the remainder and the final result of Q represents the value of the quotient.

The above is done for both DIV and DIVU operation. A table above, picture 2 shows an example of how the non-restoring division algorithm is applied. For DIV operation, the difference between that and DIVU operation is that the most significant bit (MSB) for DIV operation is taken down first and if the operands are negative value, it will be converted to its positive 2's complement form. Only the positive value of the operands will be taken into the calculation in the non-restoring division algorithm. After the whole process of non-restoring division algorithm is applied, there will be a final comparison between the dividend and the divisor, and the dividend and the remainder. If the MSB of the dividend and the divisor are different, this means that quotient is a negative value, thus we have to convert it to a negative value by changing the initial quotient value to its 2's complement form. Similarly if the MSB of the dividend and the remainder are different, it means that the remainder should be a negative value thus, the remainder has to be converted into a negative value by changing the initial remainder into its 2's complement form.

3: Total Resource & Usage Maximum frequency of operation

Timing Summary

Minimum period: 8.580ns (Maximum Frequency: 116.556MHz)

Minimum input arrival time before clock: 11.974ns

Maximum output required time after clock: 4.413ns

Maximum combinational path delay: 6.967ns

Device Utilization Summary

Selected Device: 7a100tcsg324-1

Slice Logic Utilization:		
Number of Slice Registers:	454 out of 126800	0%
Number of Slice LUTs:	2143 out of 63400	3%
Number used as Logic:	2143 out of 63400	3%
Slice Logic Distribution:		
Number of LUT Flip Flop pairs used:	2179	
Number with an unused Flip Flop:	1725 out of 2179	79%
Number with an unused LUT:	36 out of 2179	1%
Number of fully used LUT-FF pairs:	418 out of 2179	19%

Number of unique control sets:	9

IO Utilization:

Number of IOs:	170		
Number of bonded IOBs:	170 out of	210	80%

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs:	1 out of	32	3%