Computer Arithmetic Assignment,Karatsuba Multiplication of big and small numbers

Teacher: Professor Lambert Spaanenburg

By: Peyman Pouyan

Part1: Multiplication of small numbers:

We have two numbers in format of:

$A=a_kB^m+a_r$

$B=b_jB^m+b_r$

If the base is 2 then we can write the numbers in format of:

A=2^j+Ar ,

B=2^k+Br

Then the result of karatsuba multiplication in base 2 is : {2^(j+k) + Br*A + Ar*2^k}. We can see that the first and last terms are only shift operations, but to compute the middle term (Br*A) we need to call the function again, up to the time that Br or A becomes zero. So we will have a recursive function to do this.

The C code of this part is in attachments.

Part2: Multiplication of Big numbers

Constructing big numbers:

I make my big numbers by making them in format of table of 8bit arrays. It is possible to define the size of this table in the program.

	$a1_7$	$a1_6$	$a1_5$	$a1_4$	$a1_3$	$a1_2$	$a1_1$	$a1_0$		$a0_7$	$a0_6$	$a0_5$	$a0_4$	$a0_3$	$a0_2$	$a0_1$	$a0_0$
….	$a3_7$	$a3_6$	$a3_5$	$a3_4$	$a3_3$	$a3_2$	$a3_1$	$a3_0$		$a2_7$	$a2_6$	$a2_5$	$a2_4$	$a2_3$	$a2_2$	$a2_1$	$a2_0$

How does the program work for big numbers?

As shown above there are a few operations done to calculate the multiplication, these operations are :Addition, Subtraction and Shifting. So I have made functions of these operations so that they can be applied to the format of my big numbers. These functions are described below:

Addition:

When we add two numbers of width W_N and W_p the maximum width of the result will be W_n+W_p+1.So in order to be able to do addition for such a number which is located in a big table we should recognize if a carry is generated or not from addition of two 8 bit arrays, and if it is generated we propagate this carry all in the table. A way to understand this is to say if the result of addition of two 8bit arrays is more than 255 then a carry is generated.

				a[1]										a[0]				
….	$a1_7$	$a1_6$	$a1_5$	$a1_4$	$a1_3$	$a1_2$	$a1_1$	$a1_0$	+	$a0_7$	$a0_6$	$a0_5$	$a0_4$	$a0_3$	$a0_2$	$a0_1$	$a0_0$

				b[1]										b[0]				
….	$b1_7$	$b1_6$	$b1_5$	$b1_4$	$b1_3$	$b1_2$	$b1_1$	$b1_0$	+	$b0_7$	$b0_6$	$b0_5$	$b0_4$	$b0_3$	$b0_2$	$b0_1$	$b0_0$

If b[0]+a[0] >255 then we have carry and we should propagate it in table.

Subtraction:

The subtraction is also like addition and it works for subtraction of two numbers in format of big table.So we need a mechanism to recognize the carry in subtraction of two 8bit arrays and then we consider it in the whole table.

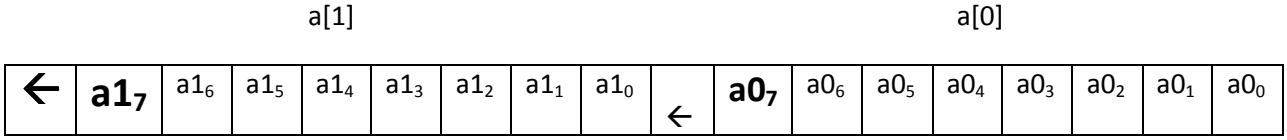
				a[1]										a[0]				
….	$a1_7$	$a1_6$	$a1_5$	$a1_4$	$a1_3$	$a1_2$	$a1_1$	$a1_0$	-	$a0_7$	$a0_6$	$a0_5$	$a0_4$	$a0_3$	$a0_2$	$a0_1$	$a0_0$

				b[1]										b[0]				
….	$b1_7$	$b1_6$	$b1_5$	$b1_4$	$b1_3$	$b1_2$	$b1_1$	$b1_0$	+	$b0_7$	$b0_6$	$b0_5$	$b0_4$	$b0_3$	$b0_2$	$b0_1$	$b0_0$

If b[0]-a[0] <0 then we have carry and we should propagate it in table.

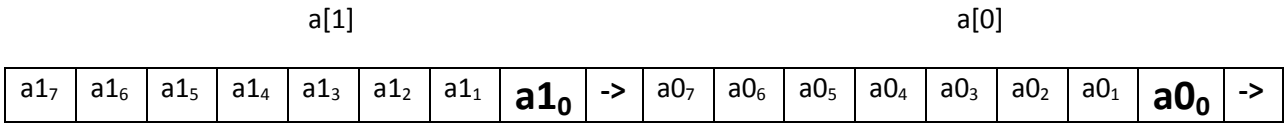
Shift left:

When we shift left a number of a format of big table then the issue is the MSB bits of the 8 bit arrays that are becoming shifted to the left, by recognizing if this number is 0 or 1 the left, shifting can be done easily.



Shift right:

While right shifting a number of a format of a big table the key point is to recognize the LSB bits of the 8 bit arrays to see whether it is 0 or 1, when this test is done right shifting will be done easily.



Initialization:

This is a function to initialize a number of a format of big table with LSB "1" and the other bits to zero.

.....	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	1

Copy:

This is for copying two numbers of format of big tables.

Iszero:

This is to recognize if a number of format of a big table is equal to zero or not.

Verification:

In order to test to see if my big multiplication is working or not I have used GMP.GMP is a free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers. The main target applications for GMP are cryptography applications and research, Internet security applications, algebra systems, computational algebra research, etc.

It is distributed under the GNU LGPL so This license makes the library free to use.

So for verification of multiplication of big numbers I do the multiplication with GMP and my own algorithm.

Results:

I was able to do multiplication for very big numbers; I tried up to big numbers, as big as the result was 1200 digits.

I did not go further but it was working further also.

I think the limitation of working of my multiplication algorithm is the limitation of memory.

The multiplication done by GMP was almost instantaneous but the multiplication by my algorithm was taking different seconds for different size of big numbers.

I generate the big random numbers in hexadecimal then I do the multiplication and show the result in hexadecimal also.

Here are some results:

Result=240digits

Number1=000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000009158427178F3FEEAC2E9B71819613E8A080D6BA41D
D1C3968F4DF6B5C2780405090639ECA9E71410E4EB559F51260B567DC2

Number2=000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000E24A7EE3971BBB868046650F1701020C1275AB56BD
6D7EA35DDAA01717552C155DB6AE379F2D66BCEE35DF609663A74F5BCF

GMP Result:
807A36360955EFB10DA0B1DF4254F71C01932A14078C929A01CA787058C56F9C292EE08
A4033EDBB67671C6CC1805799662B3C4646751FF1707D0326A68C6FAADD9498E2F554E
FE08FAB08CCC198805822D4A6B8D2889947D51AC553F0B05D8B4681A5DE

My algorithm result:
807A36360955EFB10DA0B1DF4254F71C01932A14078C929A01CA787058C56F9C292EE08
A4033EDBB67671C6CC1805799662B3C4646751FF1707D0326A68C6FAADD9498E2F554E
FE08FAB08CCC198805822D4A6B8D2889947D51AC553F0B05D8B4681A5DE

Process Time:0m0.466s

Result=722 digits

Number1=0000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000BF
C1FC4F8D1680ACD2A6F6A5DEC7B7DDE13A2485732E5104122F13238ADA65FBECDBE
884A2D291CFA1E8DA4C1B86E1A5F5EDF0CBFA84AE1FF09F88822C40288D597FFFCBD
46B1416BD4F12F640D9022BF5D674056BE21DCE4E5CD9ECE252A3E2C28AE0F97428670
F5C602F709E6732DCAA9F004CC5A33DA11BA60D322B60AE8A161F071F52C4A99FA764
E701617EB01ECB1F6B9B3558

Number2=0000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000009E
DD89B19BF3DA23AA08D4F1EA41B7463510605CB2F3D6A3FEB40B64278081CD80DE71
929D8B534F07CAECA3080DE51D7A494BF94F828A04B545D23F3668665BAAF4A149707
FF8DA1E60521AC6E47D98B8368F4BDC91F6B56B354C11C34149F4668A9874742DB2990
14A16AB995E3C5829166CCCC28F66C445B49612610F09C63ED23EBD5FDEF7ED8593AC
23229970B8D862A7405AD7

GMPresult:76FFAB53C3B7D1A5E84E92FF1793FDAD6493EF984A943EE1D2A8697B9CC7
90DC9D0A7F9845B01393D5BA2D9ACE534063FF1987EF98869002006C0B0872AA5400A0
54497921B23FD694F8C2AF39D72BBBDCC24E9B0BCAD5410793DCC71A4D974992603A
3992ECD11D446072BEAD79BCA09BFBC845512E0251881A8781DF5617FA1C1B6F768A9
26D1E0AB7210332A2F2A5345162DD6DCF743A576303D69FE5928D5888301500514F4173
86A14F57BAADBE4E75340C49B170A3A678297725B29872C122A5D389FC3C5CC098CF4
29D3DB5C5E837DAB36DDBBD50DAAF6651D9495E0C691CF154704C590C46818E3549F8

270F9AF81D6D4B706401BD0FB3A6D31B5FC773448CA7A53785940B3C4FAEE2DACAF1
0975D0D7A57D8310DF132C4D4D08E5A8FFB59ED401FF7581AE1ABCE8

Myresult:76FFAB53C3B7D1A5E84E92FF1793FDAD6493EF984A943EE1D2A8697B9CC790
DC9D0A7F9845B01393D5BA2D9ACE534063FF1987EF98869002006C0B0872AA5400A054
497921B23FD694F8C2AF39D72BBBDCC24E9B0BCAD5410793DCC71A4D974992603A39
92ECD11D446072BEAD79BCA09BFBC845512E0251881A8781DF5617FA1C1B6F768A926
D1E0AB7210332A2F2A5345162DD6DCF743A576303D69FE5928D5888301500514F417386
A14F57BAADBE4E75340C49B170A3A678297725B29872C122A5D389FC3C5CC098CF429
D3DB5C5E837DAB36DDBBD50DAAF6651D9495E0C691CF154704C590C46818E3549F827
0F9AF81D6D4B706401BD0FB3A6D31B5FC773448CA7A53785940B3C4FAEE2DACAF109
75D0D7A57D8310DF132C4D4D08E5A8FFB59ED401FF7581AE1ABCE8

Process Time:0m10.649s

Result=1204 digits

Number1=000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000001D1857C327BAC2BD600041CC4DDC1870
46285E5A1C5FE4AA769DB08CC6E3B20934E7C9AA916D8F5A514D271632DDA91D7AA
A5497CB92575088A076025C904B35AE2D85116DA123FB2F45B4F4CB946C89D49E6538C
4C17CFADBFD20FE0A0C0860F47365F079E890403A237429B00D3AB40FC3CAF9C5FF66
960BC0958FA32BDC7D668BD53BE13FB58F4948A9A5457D1D005355CFB089876FEBCB1
F272C8A5101810A36734417954F65B4D473A227C51B37E5C011731C865CF7AE91DC7D10
5D4345FBF7BBA6298C94EFF4B3673559D02E832D3EC40CAEEC33CBB315882A054530D
84617B562324B22954A42FC4B434033F32614744CF4F4AF0417AC

Number2=000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000889DEB8515DE1D45AAB869FE68DE53F8
329242E50A751705F702D9EDFAE4E6DD044354E5045AEDA89DAFB6A8DF06ADC6DF02
C450A4C4F1AB9BA548108417CA1AB659B0D3D9E7BCAB408FF1EB927695DCFE5AAC2
AB1F5E36B3D5A217E36B86D2F2874B443E1553FA0BA7FEA459CDDE0EB4030849B3A48
D6C050DED74CD022FA848B2890B1EF3E16B83474C46DA9A75154D6DAE6FCCB87811B
84DB00FDF42847266D580429086CEB10CB223303432B9429060D6AD82C85844380D03077
929C152A7BE8E5ED9843DB277E1A4AE1FA33C85D160A23634E6516D5AAD26C0190731
8790ADB76A6981689F94613F7B7EDCB26FDD0701D4BA952DC81528A

GMPresult:0F86E151F87C907CE855AF7AD96878E7152FBE2734DB5C3D6297B04AA1522F
ACB23F65D4A075D4B6BB471324007C60EF9050B565B64CF01C5B2BE74F7545AF7AC6A
78A7E306647614C6DFBF48740EEEF899A684F4EFE9B9E2E166175C61D60B14CD291D03
C28AE9597A3F92F369F96BE4F49E04A51E99F50FC575CAA2CD4E1A780E9C5C16C20DC
91A1E0B6451306EC28BC26FA211BC2107036E2219183DE1242769995D6D074A8442CBB4
527497A6F6E3AF6AA07B974C1C07203D2AE7A251A4C16881D737B3DE8FC10C190E085
45C9B01C655E6C2DE60BD1FBFCF3EA8693E2F7531DB0A1E6E36277642D6C96A8B3A23
5D291CB27B8D60ECA43DA23FD218732B10A10BABEB2934D78281ADCDDF31B1EAB5
E114083C86F63517DA44F7D8229DCF3C1AFD79E5365F6854CDA69B21812A913464A286
90AD4652EC608869450A0868909F70827D239A020F62A785C60C2CA0078BA90CCDA35E
ADF98D56B886184971F470D71B7443D5842111A8B80AA8A51F83CE9AD8359E4CD5C8E
9B3FD7ABB480336A690E99464BD7180B95F2F385A121ECBDF6C490DFC782B504DE76F
7E6C033F4FD601B4DDEF7786985FA1F7EF0B1A6250F46DFDE8D3D2A1AA2AEB42F5EA
279773EABBC4A079E1ADB87D99841D44E7BFA1723BE6EC9C333EAD2693EC1DF958E3
EB6793D7E0FE6FE520366A79C3BFA42756575DAB8

Myalgorithmresult:0F86E151F87C907CE855AF7AD96878E7152FBE2734DB5C3D6297B04A
A1522FACB23F65D4A075D4B6BB471324007C60EF9050B565B64CF01C5B2BE74F7545AF
7AC6A78A7E306647614C6DFBF48740EEEF899A684F4EFE9B9E2E166175C61D60B14CD2
91D03C28AE9597A3F92F369F96BE4F49E04A51E99F50FC575CAA2CD4E1A780E9C5C16
C20DC91A1E0B6451306EC28BC26FA211BC2107036E2219183DE1242769995D6D074A844
2CBB4527497A6F6E3AF6AA07B974C1C07203D2AE7A251A4C16881D737B3DE8FC10C19
0E08545C9B01C655E6C2DE60BD1FBFCF3EA8693E2F7531DB0A1E6E36277642D6C96A8
B3A235D291CB27B8D60ECA43DA23FD218732B10A10BABEB2934D78281ADCDDF31B1
EAB5E114083C86F63517DA44F7D8229DCF3C1AFD79E5365F6854CDA69B21812A913464
A28690AD4652EC608869450A0868909F70827D239A020F62A785C60C2CA0078BA90CCD
A35EADF98D56B886184971F470D71B7443D5842111A8B80AA8A51F83CE9AD8359E4CD
5C8E9B3FD7ABB480336A690E99464BD7180B95F2F385A121ECBDF6C490DFC782B504D
E76F7E6C033F4FD601B4DDEF7786985FA1F7EF0B1A6250F46DFDE8D3D2A1AA2AEB42
F5EA279773EABBC4A079E1ADB87D99841D44E7BFA1723BE6EC9C333EAD2693EC1DF
958E3EB6793D7E0FE6FE520366A79C3BFA42756575DAB8

Process Time :0m49.404s

References:

Course notes of Computer Arithmetic , Lambert Spaanenburg ,2008

http://gmplib.org/