EE 675: Microprocessor Applications in Power Electronics - Assignment 1

Jayesh Choudhary 170070038

1 q15inv

$$\frac{1}{1-x} = \Sigma x^n \tag{1}$$

1.1 Approach

Since the denominator has x which goes from -1 to 1, the function goes from 0.5 to ∞. the Q15 representation works fine for negative values, but for positive values, it can go upto ∞. We are only interested in the first 10 terms of the sequence, so we can say that each term, when represented in Q15 representation, can go upto maximum of 1 making the upper limit of the function 10. We can easily scale down the function by 16 in case of positive values (since shifting 4 bits is easier) and let it be for the negative values. We use 1 as the first term and and then we move into the loop, thus dealing with the case of x = 0 as well. Since we cannot represent 1, we use 0x7fff and after the loop, we add 0x0001 to get the effect of addition with 1.

1.2 Algorithm

0x8000 is compared with AH and HI bit is set if 0x8000 is greater, i.e. the number is positive. Then corresponding changes are made.
For negative value VarC = 1 + x (calculated initially)
loop (8):
VarB = x*VarB (VarB turns into x, x^2, x^3 and so on)
VarC = VarC + VarB (We keep on adding it to VarC)
VarC contains the final answer
For positive value VarC = (1 + x)/16 (calculated initially)
VarB = x/16 loop (8):
VarB = x*VarB (VarB turns into $x^2/16$, $x^3/16$, $x^4/16$ and so on)
VarC = VarC + VarB (VarC is accumulation of $1/16$, $x/16$, $x^2/16$, $x^3/16$, $x^4/16$ and so on) VarC contains the final answer scaled by 16

2 q15exp

$$e^x = \Sigma \frac{x^n}{n!} \tag{2}$$

The recursion is pretty similar to that of the last question. We just keep on dividing the VarB by an increasing sequence to get the factorial. The signed division is done using repetitive subtraction The range of this function is 1/e to e. That is why we can scale the solution by 4 for positive input and let it be same for the negative input.
VarC = 1 + x (calculated initially) and AR2=1
loop (8):
AR2++
VarB = (x/AR2)*VarB (VarB turns from x into $x^2/2$, $x^3/6$, $x^4/24$ and so on)
VarC = VarC + VarB (We keep on adding it to VarC)
VarC contains the final answer
For positive values, we scale the initial value to (1+x)/4 and VarB as x/4 ot get the final result scaled by 4.

3 q11mpy

$$z = x * y \tag{3}$$

We are given left justified Q11 numbers. Here the input x and y are treated as Q15 number. Since they are left justified, their value remains same in both representation. Now we perform multiplication on 2 Q15 numbers to get Q31 number. On shifting it left and taking the upper 16 bits, we get the Q15 product. Now to get the Q11 representation, we take the first 12 bits by using BIT-WISE AND operator on z with 0xfff0 to get a left justified Q11 number as a result.

4 Results and Simulation

Routine	AH	AL	Scaling	μC Output	Output	Expected
q15inv	0xc120	0x4120		0x55c2		
	-0.491211	**0.508789**	1	**0.669983**	**0.669983**	**0.6705959**
q11mpy				0xe000		
			1	**-0.25000**	**-0.25000**	**-0.25000**
q15exp				0x4e52		
			1	**0.611877**	**0.611877**	**0.6118849**
q15inv	0x4570	0x3800		0x1169		
	0.542480	**0.437500**	x16	**0.136017**	**2.176272**	**2.185696**
q11mpy				0x1e60		
			1	**0.237305**	**0.237305**	**0.237305**
q15exp				0x370a		
			x4	**0.429993**	**1.719972**	**1.720818**
q15inv	0xe000	0xc200		0x6666		
	-0.25000	**-0.484375**	1	**0.799988**	**0.799988**	**0.800000**
q11mpy				0x0f80		
			1	**0.121094**	**0.121094**	**0.121094**
q15exp				0x63b0		
			1	**0.778809**	**0.778809**	**0.778800**
q15inv	0x7ff0	0xe760		0x4fd3		
	0.999512	**-0.192383**	x16	**0.623627**	**9.978032**	**2049.180**
q11mpy				0xe760		
			1	**-0.192383**	**-0.192383**	**-0.192383**
q15exp				0x56ee		
			x4	**0.679138**	**2.716552**	**2.7169556**

Table 1: All possible sign combination of AL and AH

In these observations we see that for +ve values, the microcontroller output of q15inv and q15exp are scaled version of the desired output and we need to multiply the μC output with the Scaling given to get the expected output.
Here we can see that apart from the q15inv of the very high value, all obtained results are very close to the corresponding expression. This inconsistency is due to the fact that the range of function is infinity but the maximum output we can obtain from 10 terms is upper-bounded by 10. Thus the maximum value that can be represented even for q15inv is the value corresponding to 10*(0x7fff) i.e. x = 0.8998 or 0x732d. For this x, we get q15inv output as 0x33fb or 0.406097 which corresponds to 6.497552 after scaling. This discrepancy is due to the fact that we are taking only 10 terns of the series.
The asm file is properly commented to understand the steps