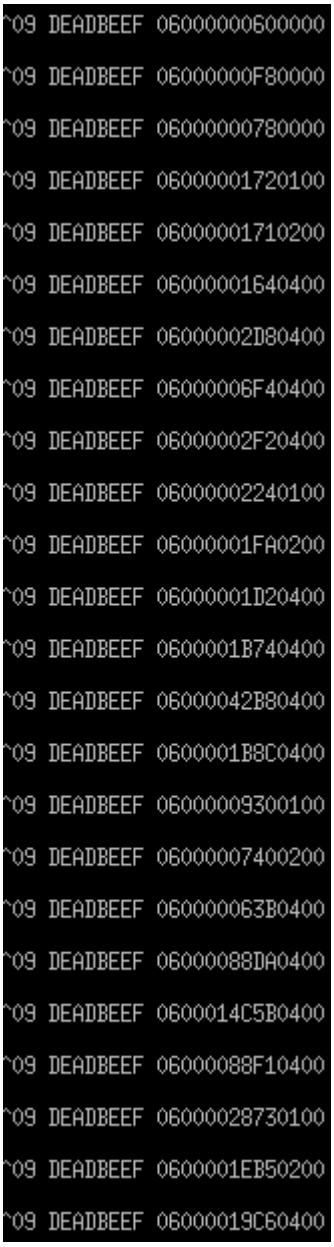
Screen shot for memset result (KL25z)



Interpretation of memset results (KL25z)

	Library function	my_memset		Dma memset		
		-O0	-O3	1byte	2byte	4byte
10	96	248	120	370	369	356
100	728	1780	754	548	506	466
1000	7028	17080	7052	2352	1856	1595
5000	35034	85083	35057	10355	7861	6598

Screenshot of memmove result (KL25Z)

Interpretation of memmove result (KL25z)

	Library function	my_memset		Dma memset		
		-O0	**-O3**	**1byte**	**2byte**	**4byte**
10	137	309	155	412	413	398
100	947	2199	965	592	548	510
1000	9050	21101	9065	2394	1900	1637
5000	45053	105104	45068	10397	7903	6640

Conclusions from profiler run on KL25z

- All results in the above table are in clock cycles.
- It can clearly be seen that DMA transfers pay off only for a large number of bytes.
- Library routines written by manufacturer (ARM) are highly optimized
- The functions written for the project (my_memset & my_memmove) perform very poorly without optimization and with maximum optimization come close to library routines
- DMA because it uses hardware for transfers is way faster than any software written transfers. Also DMA efficiency depends on byte size. Using the lasrgest byte size that DMA supports (the CPU bus size) results in the most optmized transfer