Optimization of Advanced Encryption Standard (AES) in Software

Lab#5 Report

Pratheep Joe Siluvai (pi4810)
CMPE.661 HW SW Co-design for Crypto Apps
April 3, 2014

Instructors: Kurdziel, Michael & Lukowiak, Marcin
T.A: Skalicky, Sam

Introduction

This lab was to optimize the software implementation of Advanced Encryption Standard to improve performance by different source code manipulation techniques for targeting the MicroBlaze processor.

Optimization Techniques

1. The first optimization technique is to unroll the loops in the hotspots identified from the gprof. Hence for this optimization the SubBytes operation loop is unrolled and then profiling is done.
2. The second optimization technique is to use look-up-tables (LUTs) to reduce the amount of computation performed during runtime. Since the MixColumns operation is biggest bottleneck of the algorithm, we replace the multiplication operation with pre calculated multiplication values as an LUT.
3. The third optimization technique is to use function inlining that reduces the overhead and excess instructions fetched while invoking functions.
4. The fourth optimization technique is to use different sized data structure for storing key schedule, plain text, and expanded key and state array variables in the implementation. This technique reduces the overhead of manipulating single bytes instead of entire 32-bit words of data.

Comparison of different optimization technique on performance and code size

The Table1 & Table2 shows the different optimizations and its performance matrices for different functions based on gprof and the hardware profiling and the code size for each optimizations respectively. Gprof is done for 100 loops while the hardware profiling is done only for a single run. The values for the Initialization parameter is zero because the init_param function uses very less time for initialization. The discussion about different optimization techniques are shown below.

- The Opt-1 when compared with the original implementation, the code size increases due to the unrolling but the performance improves for SubBytes calculation since the loop is unrolled for the SubBytes while it loops in original implementation. It is also noted the the Cache configuration performs better.
- The Opt-2 when compared with the original implementation, the code size and data size decreases as we are replacing the computations with the look-up-tables for the MixColumns. The performance is also improved which is shown in the gprof of Mix Columns section. Again the cache configuration continues to perform better.
- The Opt-3 when compared to the original implementation, the code size is not much decreased due to the inlining of functions. There is not much improvements in the performance with and without cache.
- The Opt-4 when compared to the original implementation, the code size and data size reduces size we are replacing all bytes to 32-bit words. The performance has also shown a reasonable improvements on cache configuration.

As a result in order to obtain better optimization on the software implementation of the AES, the software program has to combine all these four optimization in a single software implementation of AES.

Table1: Comparison of gprof and hardware profiling

Functions of Aes_128.c	Timing values using gprof									
	With Cache					Without Cache				
	Original AES	**Opt 1**	**Opt 2**	**Opt 3**	**Opt 4**	**Original AES**	**Opt 1**	**Opt 2**	**Opt 3**	**Opt 4**
AddRoundKey	1.356ms	839.363us	1.38ms	1.363ms	657.9us	1.23ms	995.363us	1.15ms	1.1ms	627.2us
ExpandKey	18.5ms	18.300ms	24.599ms	24.299ms	18ms	13.69ms	16.299ms	41.0ms	13.499ms	26.699ms
MixColumns	8.234ms	7.704ms	4.763ms	8.246ms	8.614ms	8.39ms	8.655ms	5.235ms	8.34ms	8.9ms
ShiftRows	689.5us	873.99us	696.0us	693.8us		839.5us	1.0ms	556.699us	828.8us	
SubBytes	822.2us	374.200us	745.3us	820.4us	369.2us	744.2us	417.199us	819.68us	742.29us	450.6us
Hardware Counter – Profiling										
	No. of Cycles (with cache)					**No. of Cycles (without cache)**				
	Original AES	**Opt 1**	**Opt 2**	**Opt 3**	**Opt 4**	**Original AES**	**Opt 1**	**Opt 2**	**Opt 3**	**Opt 4**
Initialization routine	0	0	0	0	0		0	0	0	0
ExpandKey	160376	160376	160483	160413	134715	160356	160305	160398	160305	134756
AddRoundKey	21812	21833	21812	21812	13528	21812	21799	21812	21812	13528
SubBytes	17739	6097	17738	17739	9352	177756	6115	17716	17756	9355
ShiftRows	4642	4641	4654	4655		4676	4651	4655	4676	
MixColumns	41524	41502	23643	41509	37984	41534	41539	23660	41524	37996
Average Round Time	78538	66914	62471	78541	54893	78544	66919	62473	78550	54890
Total Encryption	6379491	6379522	6379377	6379420	6379454	6379420	6379497	6379488	6379423	6379426

Table2: Comparison of Code Size and Data Size

Implementation	Text Size	Data Size	Bss Size
Original AES	28892	1412	2696
Opt 1	29236	1412	2696
Opt 2	12302	388	2544
Opt 3	28892	1412	2696
Opt 4	12622	388	2544