PRINCIPLES OF EMBEDDED SOFTWARE

MEMORANDUM (REPORT), ASSIGNMENT – 5

Function Table:

(w/o T) represents the value when run in absence of (without) test suit.

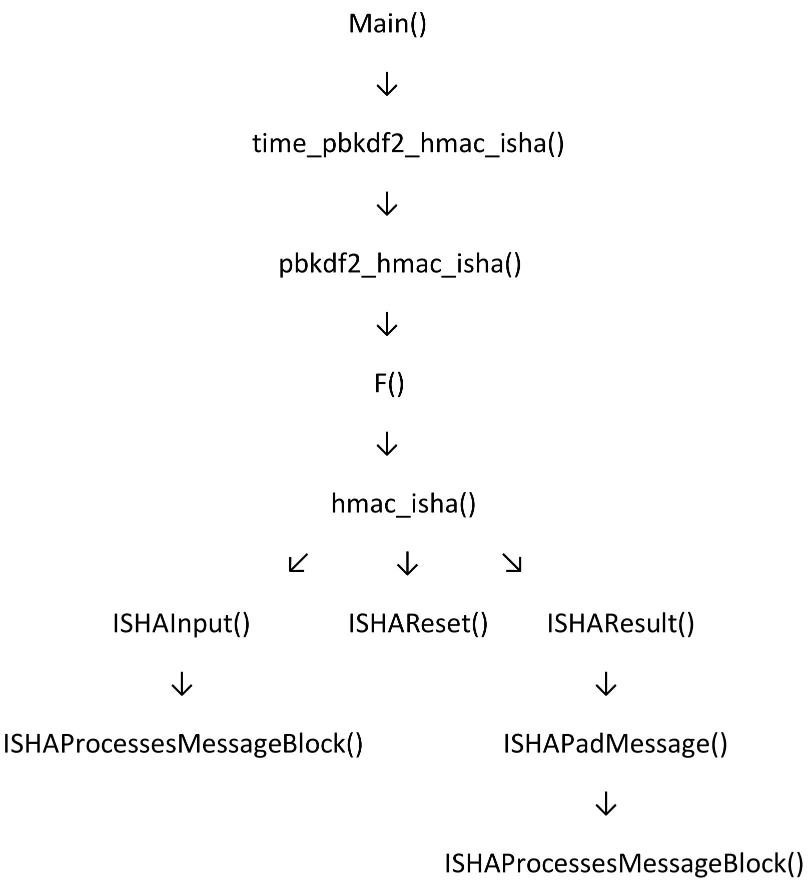
(w T) represents the value when run in presence of (with) test suit.

FUNCTION	NO. OF CALLS	OVERALL TIME TAKEN(ms)
pbkdf2_hmac_isha()	1(w/o T), 12(w T)	8744(w/o T), 9480(w T)
hmac_isha()	12288(w/o T), 13330(w T)	8601.6(w/o T), 9164(w T)
F()	3(w/o T), 42(w T)	8742.6(w/o T), 9479(w T)
ISHAProcessMessageBlock()	49152(w/o T), 53336 (w T)	2838(w/o T), 3101(w T)
ISHAPadMessage()	24576(w/o T), 26668 (w T)	1895(w/o T), 2071(w T)
ISHAReset()	24576(w/o T), 26668 (w T)	66(w/o T), 71 (w T)
ISHAResult()	24576(w/o T), 26668(w T)	2151.6(w/o T), 2372 (w T)
ISHAInput()	49152(w/o T), 53356(w T)	4227.1(w/o T), 5238 (w T)

Full Call Stack for functions:

The program starts by calling main(). The function main() calls time_pbkdf2_hmac_isha() which in turn calls pbkdf2_hmac_isha(). Then from pbkdf2_hmac_isha(), the function F() is called. F() then calls hmac_isha(). Now, hmac_isha() calls three functions which are - ISHAReset(), ISHAInput(), and ISHAResult. From ISHAInput(), ISHAProcessesMessageBlock() is called. While from ISHAResult(), ISHAPadMessage() function is called which in turn calls ISHAProcessMessageBlock(). ISHAReset does not call any function.

Below is the flow diagram where upper function calls the lower function.



As I analyzed through the above table of number of calls of and time taken by each function and noticed the call flow, The function ISHAProcessMessageBlock(), ISHAPadMessage(), and ISHAInput() were the key locations to optimize. Though the single call of these functions took negligible amount of time, they were put inside large loops and hence the overall time they took was substantial enough to optimize. Due to occurrence of these functions in loops, even small logical change reduced run time substantially. The approach I took to analyze the program was mixed with commenting out functions (blocks), returning from the function at the start of its definition, using systick get_timer() to calculate single execution time, and using variable incrementation to find out number of calls for each function. Loops, I think, were the key areas too to optimize as the iterations of some loops were quite large.