CS 577: Project Report

Group Name	VLSI_005
Top Module	Crypto_kem_dec

Group Members	**Roll Numbers**
Ashish Kumar Pal	224101009
Kaja Gnana Prakash	224101027
Karthik Maddala	224101029
Tejas Chandra Karredula	224101052
Vudatha venkata Narendra	224101060

Initially the given project files are not getting synthesized and not able to run co simulation

Changes done in order to make it synthesize:

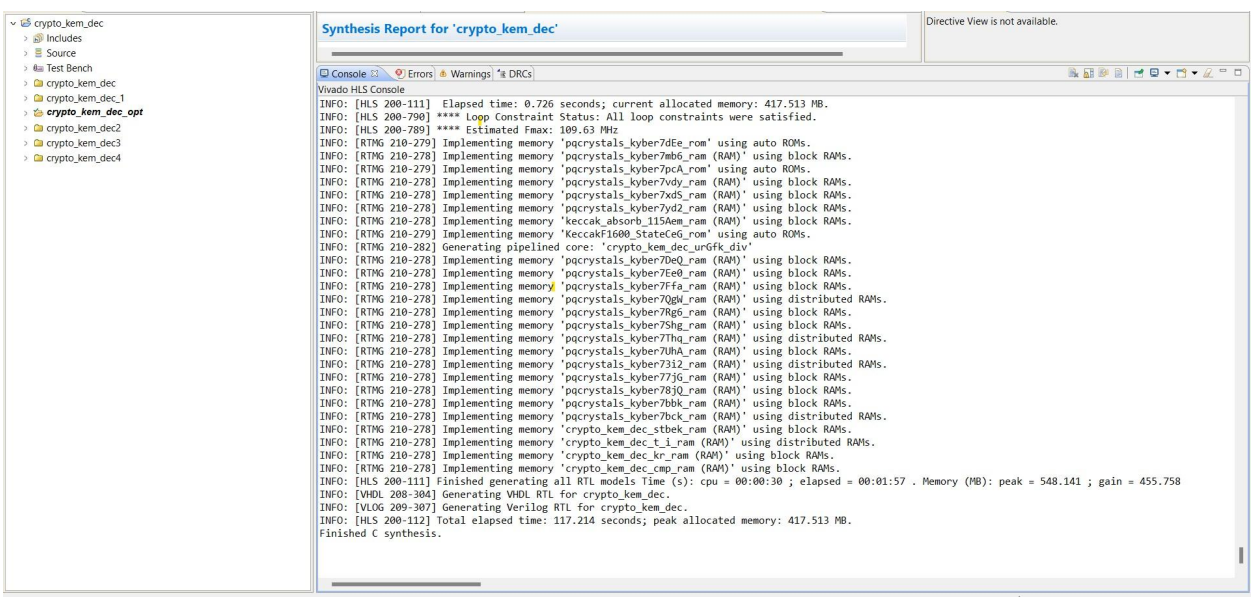
1)Comment out the kem.h header in the kem.c file.

We have done this change because all function call definition in kem.h are in the pointer format which are not compatible.

2)Excluded speedprint.h, testspeed.c from project workspace.

We have done this because these particular files have no significance with respect to our top function.Therefore these files can be excluded.

Screenshot:



Changes done in order to make it co-simulate:

1)In the top module for parameters included the array range ss[CRYPTO_BYTES],ct[CRYPTO_CIPHERTEXTBYTES],sk[CRYPTO_SECRETKEYBYTES].

We have done this change because, as pointers are not supported we made all of them static and passed the argumented based on their limits.

2)As these are present in “api.h” header, include “api.h” in “kem.c” file.

We have done this change because api.h is the file that contains all preprocessors that we passed to our static arrays(ss,ct,sk).

```
Vivado HLS Console

source C:/Users/kmadd/AppData/Roaming/Xilinx/Vivado/crypto_kem_dec/crypto_kem_dec_opt/sim/verilog/xsim.dir/crypto_kem_dec/webtalk/xsim_webtalk.tcl -notrace
INFO: [Common 17-186] 'C:/Users/kmadd/AppData/Roaming/Xilinx/Vivado/crypto_kem_dec/crypto_kem_dec_opt/sim/verilog/xsim.dir/crypto_kem_dec/webtalk/usage_stat
webtalk_transmit: Time (s): cpu = 00:00:00 ; elapsed = 00:00:06 . Memory (MB): peak = 109.883 ; gain = 17.578
INFO: [Common 17-206] Exiting Webtalk at Fri Mar 10 10:29:09 2023...

****** xsim v2019.2 (64-bit)
  **** SW Build 2708876 on Wed Nov  6 21:40:23 MST 2019
  **** IP Build 2700528 on Thu Nov  7 00:09:20 MST 2019
    ** Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.

source xsim.dir/crypto_kem_dec/xsim_script.tcl
# xsim {crypto_kem_dec} -autoloadwcfg -tclbatch {crypto_kem_dec.tcl}
Vivado Simulator 2019.2
Time resolution is 1 ps
source crypto_kem_dec.tcl
## run all
////////////////////////////////////////////////////////////////////////////////
// Inter-Transaction Progress: Completed Transaction / Total Transaction
// Intra-Transaction Progress: Measured Latency / Latency Estimation * 100%
//
// RTL Simulation : "Inter-Transaction Progress" ["Intra-Transaction Progress"] @ "Simulation Time"
////////////////////////////////////////////////////////////////////////////////
// RTL Simulation : 0 / 2 [n/a] @ "125000"
// RTL Simulation : 1 / 2 [n/a] @ "1862065000"
// RTL Simulation : 2 / 2 [n/a] @ "3723515000"
////////////////////////////////////////////////////////////////////////////////
$finish called at time : 3723555 ns : File "C:/Users/kmadd/AppData/Roaming/Xilinx/Vivado/crypto_kem_dec/crypto_kem_dec_opt/sim/verilog/crypto_kem_dec.autotb
run: Time (s): cpu = 00:00:00 ; elapsed = 00:00:31 . Memory (MB): peak = 240.367 ; gain = 0.000
## quit
INFO: [Common 17-206] Exiting xsim at Fri Mar 10 10:29:50 2023...
INFO: [COSIM 212-316] Starting C post checking ...
INFO: [COSIM 212-1000] *** C/RTL co-simulation finished: PASS ***
Finished C/RTL cosimulation.
```

Optimizations:

The main aim of optimization is to reduce latency and area overhead.

Optimization-1(Overhead):

1)In top module we observed that buf is used in many intermediate implementations so we thought of partitioning the buf array.Hence
applied cyclic partition with factor=4 dim=1.

2)In “verify.c” we observed that the loop are running into many cycles so we thought of applying the pipelining.Hence we applied loop pipelining to both the for loops inside the verify.c

3)In “indcpa.c” we observed that, this is indirectly depends on our top function. So observed all the elements came to a conclusion that we can optimize the seed array. Hence partitioned the seed array,cyclic partition with factor=2 dim=1

Optimization 2(Latency):

1)From the synthesis console we observed that poly.c have function named “poly_basemul_montgomery” which is conceding high latency.so we thought of applying pipelining with Initiation Interval=3.

2)From the synthesis console we observed that polyvec.c have functions named “polyvec_ntt” and “polyvec_invntt_tomont” and “polyvec_decompress” which are conceding high latency.so we thought of applying pipelining with Initiation Interval=3.

We have created three solutions namely

1)Crypto_kem_dec_unoptimized : This solution is unoptimized.

Area Overhead:

BRAM_18K is 10%

DSP48E is 19%

Flipflops is 10%

LUT is 90%

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	252	-
FIFO	-	-	-	-	-
Instance	64	146	27643	120044	0
Memory	15	-	48	25	0
Multiplexer	-	-	-	1112	-
Register	-	-	150	-	-
Total	79	146	27841	121433	0
Available	730	740	269200	134600	0
Utilization (%)	10	19	10	90	0

Latency:

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	187042	187066	187090	187091	187091	187091

2)Crypto_kem_dec_overhead_opt: This solution is optimized, and in this solution we aimed on reducing area overhead

Area Overhead:

BRAM_18K is 10%

DSP48E is 19%

Flipflops is 9%

LUT is 77%

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	258	-
FIFO	-	-	-	-	-
Instance	63	146	24296	103174	0
Memory	16	-	80	17	0
Multiplexer	-	-	-	1224	-
Register	-	-	140	-	-
Total	79	146	24516	104673	0
Available	730	740	269200	134600	0
Utilization (%)	10	19	9	77	0

3)Crypto_kem_dec_latency_opt: This solution is optimized, and in this solution we aimed on reducing Latency.And reduced latency ~50%

Latency:

Result

		Latency			Interval		
RTL	Status	min	avg	max	min	avg	max
VHDL	NA	NA	NA	NA	NA	NA	NA
Verilog	Pass	97946	97970	97994	97995	97995	97995