Power-Performance Tradeoff Analysis for Zynq Platform

—

SLDM project, 2015-16

Presented By: Rajesh Kedia (2014CSZ8383)

Goal

Long Term:

- Estimate power consumption of a C function being implemented as Hardware using SDSoC environment.
- Enable faster design space exploration with power (energy) as the optimization metric.

Project Scope:

- Understand the power (and energy) consumption behavior of hardware generated by SDSoC under various configurations.
- Focus on dynamic power only.

Test Case and Configuration Details

- Matrix Multiplication testcase with following variations:
  - 32x32 square matrices, each data-type being float.
  - 64x64 square matrices, each data-type being float.
  - 32x32 square matrices, each data-type being integer.
  - 64x64 square matrices, each data-type being integer.
- Various hardware configurations were generated for each case:
  - Array partitioning varied to control number of BRAMs in which the array is split: impacts latency, resource usage and power consumption.
  - Pipeline initiation interval (II) to control the degree of pipelining: impacts latency, resource usage and power consumption.
- Zynq system on Zedboard was chosen as the target hardware.

Tools Flow

1. Create different SDSoC project with specific configuration (using pragma).
2. Synthesize and generate the bitstream using SDSoC.
3. Open the implemented design in Vivado IPI.
4. Invoke power analysis command for Vivado and dump power report.
5. Invoke report utilization command for Vivado and dump usage report.
6. Extract parameters for mmult_accel_0 and compile in the excel report.

Step-1 is manual, steps 2 till 5 are automated through scripts, step-6 is manual.

Results: 64x64 matrix, floating point

Configuration	Total Power (W)	MMULT power (W)	FF	LUT	LUT memory	BRAM	DSP	Latency (cycles)	Energy	Pipeline II achieved
array_partition=32	0	COULD NOT FIT IN FPGA					0	4679	0	1
array_partition=16	2.432	0.594	23975	24671	10287	32	160	8774	5211.756	2
array_partition=12	2.273	0.442	20132	17317	7129	22	110	12871	5688.982	3
array_partition=8	2.174	0.347	17425	14749	5452	16	80	16964	5886.508	4
array_partition=6	2.09	0.264	15719	12078	3908	12	55	25159	6641.976	6
array_partition=4	2.011	0.188	13989	10239	3020	8	40	33344	6268.672	8
array_partition=3	1.981	0.162	12282	9028	2404	12	30	45632	7392.384	11
array_partition=2	1.918	0.1	9661	6301	1526	8	20	66104	6610.4	16
array_partition=1	1.883	0.06	6860	3919	856	8	10	131624	7897.44	32

Results: 32x32 matrix, floating point

Configuration	Total Power (W)	MMULT power (W)	FF	LUT	LUT memory	BRAM	DSP	Latency (cycles)	Energy	Pipeline II achieved
array_partition=32	2.57	0.698	16058	11392	732	64	160	1319	920.662	1
array_partition=16	2.577	0.707	16089	11477	1002	32	160	1319	932.533	1
array_partition=8	2.126	0.293	12044	10138	2839	16	80	2342	686.206	2
array_partition=6	2.064	0.23	10147	7300	2062	11.5	55	3367	774.41	3
array_partition=4	2.005	0.176	8801	6298	1542	8	40	4388	772.288	4
array_partition=3	1.969	0.144	8290	5585	1175	6	30	6439	927.216	6
array_partition=2	1.909	0.087	7082	4428	801	4	20	8480	737.76	8
array_partition=1	1.876	0.054	4910	2727	437	2	10	16664	899.856	16

Results: 64x64 matrix, 32-bit integer

Configuration	Total Power (W)	MMULT power (W)	FF	LUT	LUT memory	BRAM	DSP	Latency (cycles)	Energy	Pipeline II achieved
array_partition=32	2.097	0.24	6008	2166	586	64	192	4109	986.16	1
array_partition=16	1.95	0.113	4436	1702	297	32	96	8205	927.165	2
array_partition=12	1.926	0.089	3647	1854	241	22	66	12300	1094.7	3
array_partition=8	1.897	0.064	2954	1431	152	16	48	16396	1049.344	4
array_partition=6	1.972	0.14	8147	2687	558	12	171	24591	3442.74	6
array_partition=4	1.866	0.036	2154	1134	72	8	24	32780	1180.08	8
array_partition=3	2.047	0.213	7198	2712	447	12	138	45071	9600.123	11
array_partition=2	1.852	0.023	1641	1082	36	8	12	65547	1507.581	16
array_partition=1	1.845	0.016	1140	1088	18	8	6	131083	2097.328	32

Results: 32x32 matrix, 32-bit integer

Configuration	Total Power (W)	MMULT power (W)	FF	LUT	LUT memory	BRAM	DSP	Latency (cycles)	Energy	Pipeline II achieved
array_partition=32	1.977	0.13	3064	1191	296	32	96	1036	134.68	1
array_partition=16	1.969	0.124	3072	1171	296	32	96	1036	128.464	1
array_partition=8	1.892	0.058	2299	948	151	16	48	2060	119.48	2
array_partition=6	1.883	0.053	2289	1232	154	12	39	3083	163.399	3
array_partition=4	1.863	0.035	1675	865	78	8	24	4107	143.745	4
array_partition=3	1.908	0.078	3747	1502	259	6	72	6158	480.324	6
array_partition=2	1.853	0.02	1214	794	36	4	12	8203	164.06	8
array_partition=1	1.843	0.015	824	613	18	2	6	16394	245.91	16

Special cases

Testcase	Configuration	Total Power	MMULT power	FF	LUT	LUT memory	BRAM	DSP	Latency (cycles)	Energy	Pipeline II
64x64, FL	array_partition=16	2.432	0.594	23975	24671	10287	32	160	8774	5211.756	2
32x32, FL	array_partition=16	2.577	0.707	16089	11477	1002	32	160	1319	932.533	1
64x64, INT	array_partition=16	1.95	0.113	4436	1702	297	32	96	8205	927.165	2
32x32, INT	array_partition=16	1.969	0.124	3072	1171	296	32	96	1036	128.464	1

HLS with non-uniform partitioning

Configuration	FF	LUT	BRAM	DSP	Latency per iteration (cycles)	Pipeline II achieved
array_partition=3, matrix size = 32x32	2375	1598	12	**100**	21	6
array_partition=3, matrix size = 29x29	1340	1342	12	**30**	17	5
array_partition=3, matrix size = 23x23	1088	952	12	**28**	16	4
array_partition=3, matrix size = 20x20	1669	1172	12	**42**	15	4
array_partition=3, matrix size = 31x31	2357	1158	12	**100**	20	6
array_partition=3, matrix size = 33x33	3360	1647	12	**136**	21	6
array_partition=3, matrix size = 34x34	1699	1749	12	**26**	18	6
array_partition=6, matrix size = 32x32	1665	1514	24	**54**	15	3
array_partition=2, matrix size = 27x27 (21, 22, 23, 25, 26, 27)	2218	1467	8	**~82**	21	7

Other points related to experiments

- Few experiments were done using SAIF[1] dump to get realistic power estimate. It was observed that the activity information caused a uniform 15% increase in the estimates. Hence this experiment was stopped.
  - SDSoC doesn't support SAIF dump directly. Hence, the same function was synthesized and simulated in Vivado HLS with SAIF dump enabled. Manually the hierarchical path of the mmult module was added in SAIF to match the path as in SDSoC.
- Experiments were done for varying the pipeline II, but many of the cases were similar to what varying the array partition achieves. Hence this was done only for floating point case.

[1] : SAIF stands for Switching Activity Interchange Format, used to capture signal activity for power estimation.

Summary & Conclusion

- There is a drastic drop in the power consumption from II=1 case to II=2 case.
- Owing to the above behavior, pipeline II=2 gives the best energy consumption among all other cases.
- If a configuration is needed which doesn't lead to a uniform partitioning, it may be better to synthesize for a slightly larger matrix size and fill them with dummy values.

Future Work:

- Debug the cause for the drastic dip in power for II=2 compared to II=1.
- Debug the cause for the irregular partitioning - Seems to be a tool bug. Need to report to Xilinx.
- Model the findings and learnings to be able to estimate power at high level for matrix multiplication example.

References

1. SDSoC user guide
   a. ug1027-intro-to-sdsoc
   b. ug1028-sdsoc_getting_started
2. Vivado HLS user guide
   a. ug871-vivado-high-level-synthesis-tutorial
   b. ug902-vivado-high-level-synthesis
3. Vivado user guide
   a. ug900-vivado-logic-simulation
4. Vivado power estimation user guide
   a. ug907-vivado-power-analysis-optimization

Acknowledgements

- Prof. M. Balakrishnan - for problem motivation and suggestions during the project.

- Lokesh Siddhu - for some of the discussions related to generated logic and their behavior.

Thanks

BACKUP

Tools Flow

To get activity based power:

1. Synthesize the C-code using Vivado HLS and co-simulate RTL with C testbench.
2. Now, modify Vivado simulation tcl to dump activity file in saif format.
3. Rerun the simulation to get the saif file.
4. Modify saif file to update hierarchical path as per generated design.
5. Read this along with the corresponding SDSoC design and report power.