Designing Power-Efficient & Safe Autonomous Driving Devices: Using Custom Techniques to Quickly Identify/Fix Hotspots a RTL

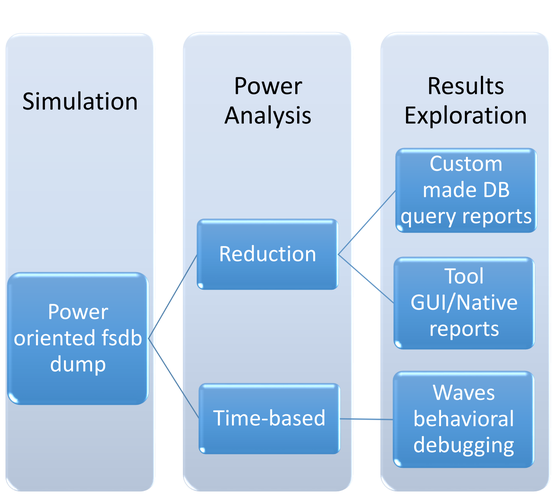
Erez Iluz

Introduction

EyeQ® family of SoCs is required to support complex and computationally intense vision processing, and still maintain **low power consumption** especially while located on the windshield.
Therefore, several high power consuming accelerators were targeted for comprehensive RTL changes in order get their power decreased significantly.

Methodology

Being developed for the automotive industry, EyeQ® development has some challenges like **Functional Safety**, **High Frequency** (2GHz), **Fast turnaround time** and more. For that, a power reduction environment at RTL stage was developed.



Waves power debugging – found redundant peak power at specific time windows

	Instance name	Total Flops	Gated Flop	CGEE %	Wasted Clock Pin
3		271	0	100.00%	-
18		504	0	100.00%	-
296		2436	2359	-	-
412		241	214	23.82%	3.19uW
454				23.61%	161mW
455		12172	12172	30.41%	23.8mW
583		24528	24528	2.41%	13.3mW
584		1533	1533	2.41%	819uW
585		1533	1533	2.41%	819uW
586		818	818	0.00%	173uW
587		149	149	100.00%	-
588		118	118	0.00%	18.7uW
589		40	40	0.00%	90.9uW
590		156	156	100.00%	-
591		41	41	100.00%	-
592		1533	1533	2.41%	832uW

CSV report helped find low-efficient CG enables with high power impact, even in lower hierarchies -> 2.41% CGEE for 13.3mW power wastage!

Using DB query based reports, we mange to pinpoint those exact desired spots of high, non optimized power consumption, address them properly and reduce power.

Flow - Power Suite

In order to ease the process of running reduction power analysis and exploring results, a power suite was built around the flow. It includes the following key features:

- Link and Compile technology libraries
- Physical calibration based on mature P&R design
- Generates design clocks list automatically
- Power oriented FSDB dump utility within testing flow

DB Query

In order to build a proper power DB, we executed several tool native commands on reduction analyses results, focusing on various CG aspects:

- CG **Coverage**
- CG **Enable** efficiency
- **Data Aware** CG efficiency

Using custom Perl scripts, we post-processed these results and exported the refined data into several Excel sheets.

With these reports, it was much easier to highlight and prioritize what exactly needs to be addressed:

Data activity vs. Clock activity:

Inst	Total Power [uW]	Total Dynamic Power [uW]	Totale Leakage Power [uW]	Register Dynamic Power [uW]	Combinational Dynamic Power [uW]	Memory Dynamic Power [uW]	Clock Dynamic Power [uW]	Clock-Activity	Data-Activity
InstA	302.362	265.469	36.893	4.617	0.216	0	260.636	0.1	0
InstB	70521.862	68286.83	2235.032	32139.48	25234.35	0	10913	0.95	0.145
InstB/InstAA	39376.194	37591.16	1785.034	9015.43	24631.67	0	3944.06	0.87	0.2
InstB/InstAA/InstAAA	18220.342	17535.125	685.217	4568.115	10994.98	0	1972.03	0.91	0.19
InstB/InstAA/InstAAA/InstABC	8724.573	8431.941	292.632	2326.086	5119.84	0	986.015	0.9	0.09

CG efficiency per Gate:

	A	B	C	D	E	F	G	H	I	J
1	Inst Name	Gater Type	Impact	En Eff	Dwnstr Power	Gated Insts	Gated Bits	En Net	File	Line
2	#m78	inferred	0.943	0	0.038		15936		File1.v	217
3	#m870	inferred	0.0234	0	0.000865		332		File2.v	320
4	#m915	inferred	0.023	0	0.00151		320		File3.v	496
5	#m880	inferred	0.0191	0	0.000556		256		File4.v	614

CG Enable efficiency per hierarchy:

Instance name	Total Flops	Gated Flops	CGEE %	Wasted Clock Pin power
InstA	287727	286513	23.61%	161mW
InstA/InstAB	12172	12172	30.41%	23.8mW
InstA/InstAC	24528	24528	2.41%	13.3mW
InstA/InstAC/InstACD	1533	1533	2.41%	819uW
InstA/InstAC/InstACD/InstACDA	1533	1533	2.41%	819uW
InstA/InstAC/InstACD/InstACDA/InstX	818	818	0.00%	173uW
InstA/InstAC/InstACD/InstACDA/InstX/InstY	149	149	100.00%	-

Non-gated/Instantiated CG for large Register bank:

	A	B	C
1	Flop	Width	CG Type
2	FlopXa[1:0][14:0][255:0]	7360	Instantiated
3	FlopYa[465:0]	466	None
4	FlopZa[1:0][4:0][15:0]	160	Instantiated
5	FlopXb[127:0]	128	Instantiated
6	FlopYb[112:0]	113	None
7	FlopsZb[3:0][15:0]	64	Instantiated

Results

After several analyses <-> RTL fix iterations, we managed to get outstanding results improving the clock gating efficiency from 50% → 80%. For example:

```
                Potential :    145mW   19.61%
are                       :
 Auto-accepted            :    110mW   14.83%
  Strengthened            :   2.62mW    0.35%
     Potential            :    114mW   15.37%
```

```
                Potential :     77.9mW  14.10%
are                       :
 Auto-accepted            :    380uW    0.07%
  Strengthened            :   1.98mW    0.36%
     Potential            :      3mW    0.54%
```