ET4171 Processor Design Project

LEON3 processor optimization

Group 1:

Luca Feltrin

Henrique Dantas

Objectives

- Target: Embedded applications
  - Compound metric: P*BS
- Poor Mul/Div execution time
  - Implementation with different algorithms

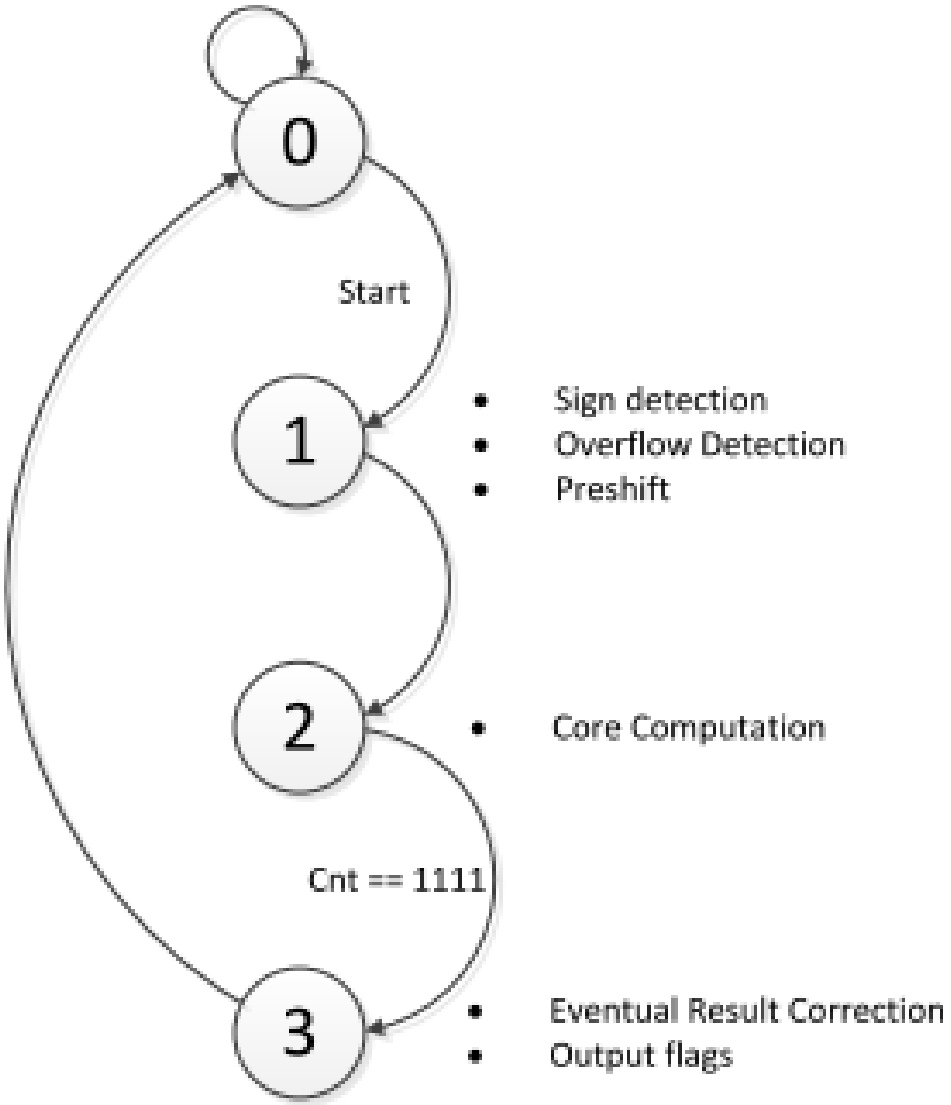
Multiplier

Divider

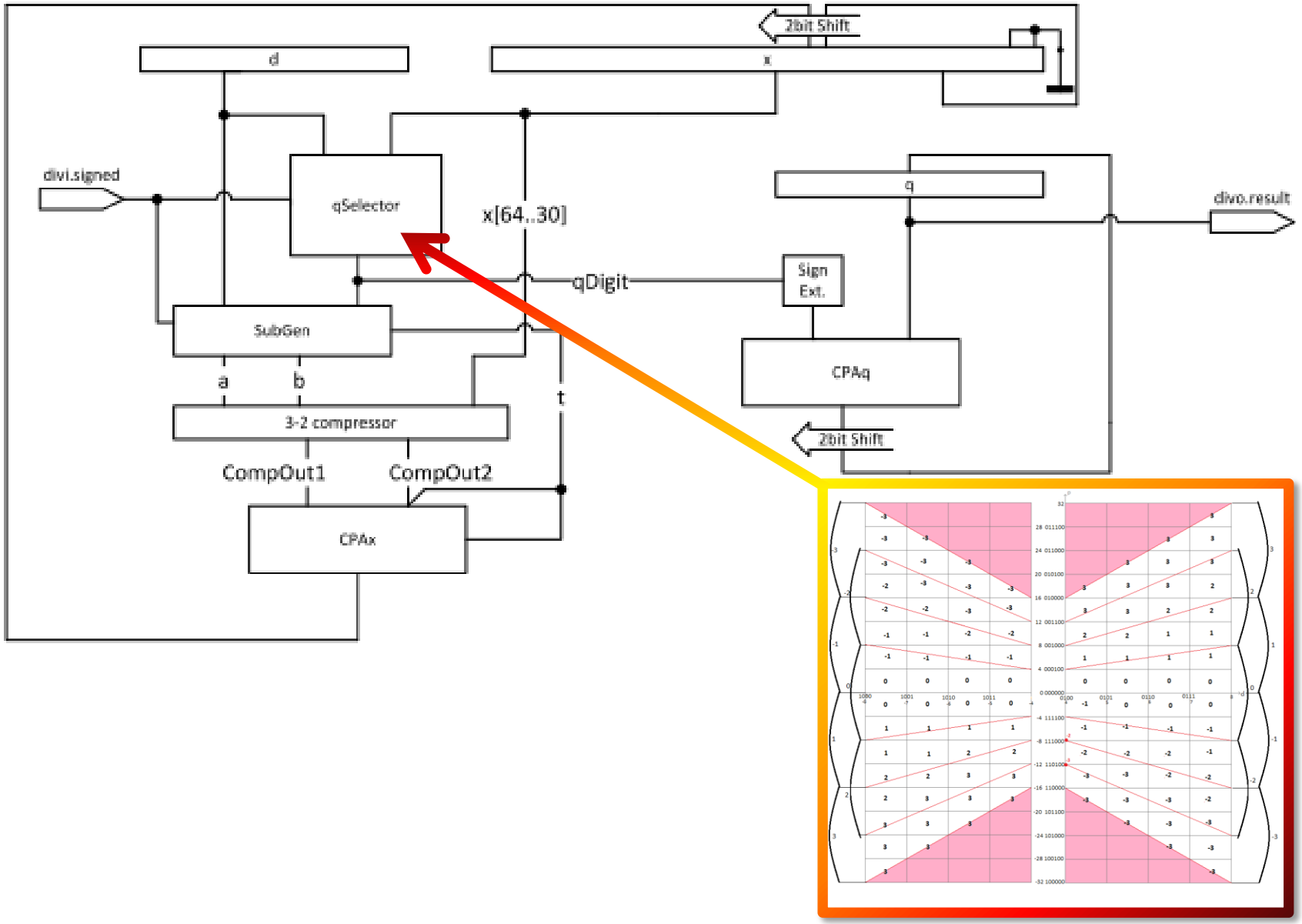
- Which algorithm?
  - Repeated Multiplication — **Fast** **Area**
  - Reciprocation — **Fast** **Area**
  - Array Divider — **No control on physical placing**
  - Radix >8 — **Fast** **Area**
  - Radix-4 — **Good compromise:**

Execution time ~ ½ of the baseline

Divider


0
Start
1
Sign detection
Overflow Detection
Preshift
2
Core Computation
Cnt == 1111
3
Eventual Result Correction
Output flags

Divider

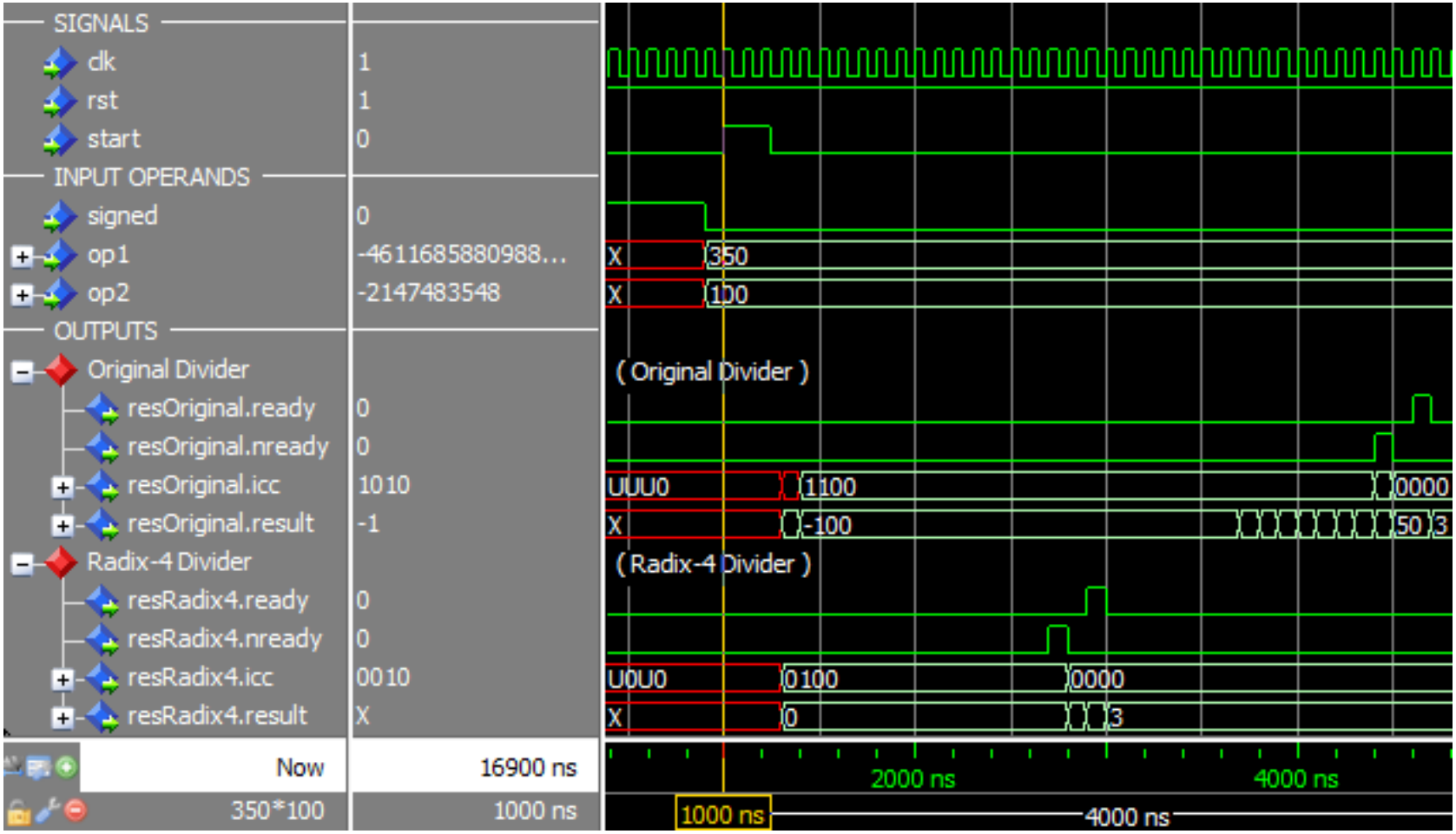

2bit Shift
d
x
divi.signed
qSelector
x[64..30]
q
divo.result
qDigit
Sign
Ext.
SubGen
CPAq
a
b
t
3-2 compressor
2bit Shift
CompOut1
CompOut2
CPAx

Divider

- Signed division (signed p-d plot) vs. Unsigned division (half p-d plot) + 1 cycle for sign
  - No area differences but 1 more cycle delay: **signed Division**

Divider

- Baseline vs. Radix-4: 19 cycles vs 36



Synthesis Results

	Clk freq [MHz]	LUTs	Slices	Quiescent Power [W]	Dynamic Power [W]	Total Power [W]	P/f [W/MHz]
Baseline	80,522	9904	16889	2,467	0,721	3,188	0,03959
Modified	80,535	10479	17865	2,468	0,743	3,211	0,03987

Benchmarks Scores

	Stanford [sec]	Whetstone [sec]	Gmpbench Multiply [Op/sec]	Gmpbench Divide [Op/sec]	Gmpbench RSA [Op/sec]	Division [sec]	Mibench JPEG (average) [sec]	SSD [sec]	Total [sec]
Baseline	2,30	116,2	781	15876	5123	8,06	23,215	10,59	**219,28**
Modified	2,26	113,25	801	16335	5284	7,65	22,465	10,21	**213,30**

Only slight improvement probably due to the Operative System's Scheduling

Conclusion

- Comparison with metrics

Version	Primitive metrics				Composite metrics			
	A (*10^4)	D (*10^-2)	P	BS (*10^2)	A*D (*10^2)	A*BS (*10^6)	P*D (*10^-2)	**P*BS (*10^2)**
Baseline	2,68	1,24	3,19	2,19	3,33	5,88	3,96	**6,99**
Modified	2,83	1,24	3,21	2,13	3,52	6,05	3,99	**6,85**
Improvements	5,8%	-	0,7%	-2,7%	5,8%	2,9%	0,7%	**-2,0%**

Furhter Improvements

- Cache size
  - More power consumption, need to determine actual miss rate
- Branch prediction
  - Now: Static prediction, only slight advantage, power consumption
- Out of Order Execution
  - Radical change of the integer unit, improved execution time