Case Study: Multi-synchronous approaches to derive energy and architectural benefits in clocked peripherals

William Lee, Vikas Vij, Dipanjan Bhadra, Kenneth Stevens

Electrical and Computer Engineering Department
University of Utah

March 28, 2013

Introduction

- Thinking out of the box – Exploit new timing models for power and/or performance improvements.
- Use the best CAD tool flows to enable the use of this new model.
- A methodology to generate multi-frequency designs to derive the benefits.
- Achieving 3× reduction in energy is common.

Introduction

- Can similar benefits be achieved on small frequency dependent peripherals like UART?
- Is it worth converting these small designs to asynchronous?
- How intrusive we need to be with respect to these modifications?

- Headsup – We achieved around 4× benefit in power for an asynchronous UART with respect to a clock gated one.

Introduction

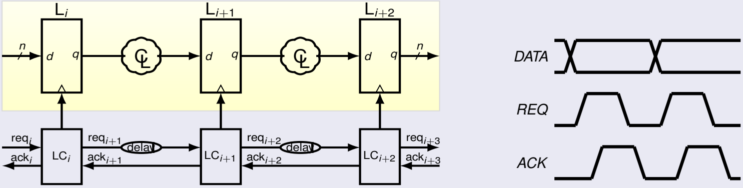
- Can similar benefits be achieved on small frequency dependent peripherals like UART?
- Is it worth converting these small designs to asynchronous?
- How intrusive we need to be with respect to these modifications?

- Headsup – We achieved around 4× benefit in power for an asynchronous UART with respect to a clock gated one.

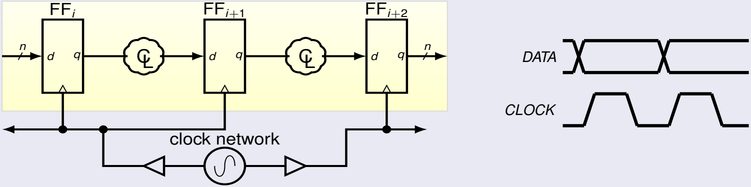
Background

- Toolflow and methodology.
- Using 64-point FFT how benefits are derived using hierarchical design and relative timing.
- Approaches used for UART and why large peripheral approaches cannot work.
- How reactive nature of the asynchronous circuits works in our favor with UART.

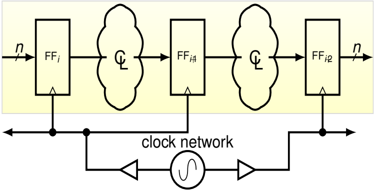
Asynchronous Design

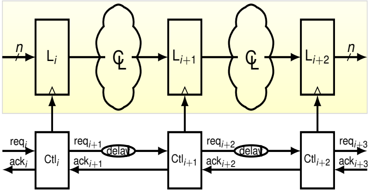


Synchronous (Clocked) Design



Relative Timing Overview

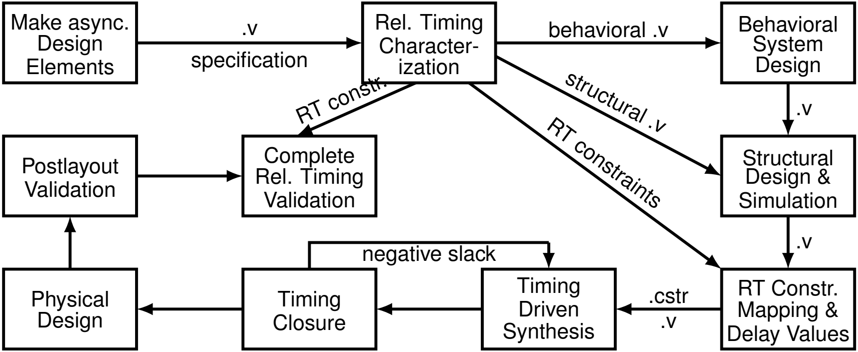




Frequency based (clocked) design. Clock frequency and datapath delay of first pipeline stage is constrained by
$\uparrow \mathrm{L}_i/\mathrm{clk}_i \mapsto \mathrm{L}_{i+1}/\mathrm{d+s} \prec \uparrow \mathrm{L}_{i+1}/\mathrm{clk}_{i+1}$

Timed (bundled data) handshake design. Delay element sized by RT constraint:
$\uparrow \mathrm{req}_i \mapsto \mathrm{L}_{i+1}/\mathrm{d+s} \prec \uparrow \mathrm{L}_{i+1}/\mathrm{clk}$

Relative Timing based Cad Tool Flow



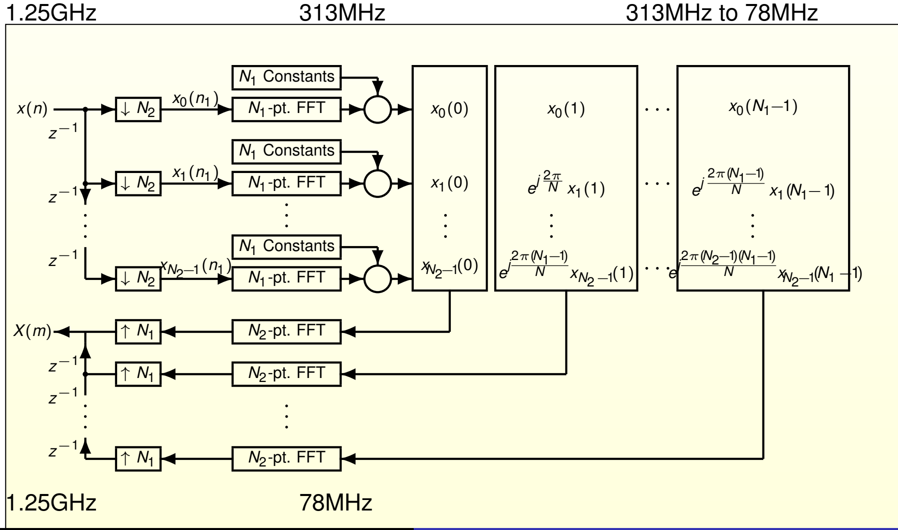
Constraints Example

```
set d0_fdel 0.600
set d0_fdel_margin [expr $d0_fdel + 0.050]
set d0_bdel 0.060

set_size_only -all_instances [find -hier cell lc1]
set_size_only -all_instances [find -hier cell lc3]
set_size_only -all_instances [find -hier cell lc4]

set_disable_timing -from A2 -to Y [find -hier cell lc1]
set_disable_timing -from B1 -to Y [find -hier cell lc1]
set_disable_timing -from A2 -to Y [find -hier cell lc3]
set_disable_timing -from B1 -to Y [find -hier cell lc3]

set_max_delay $d0_fdel -from a -to l0/d
set_max_delay $d0_fdel -from b -to l0/d
set_min_delay $d0_fdel_margin -from lr -to l0/clk
set_max_delay $d0_bdel -from lr -to la
#margin 0.050 -from a -to l0/d -from lr -to l0/clk
#margin 0.050 -from b -to l0/d -from lr -to l0/clk
```

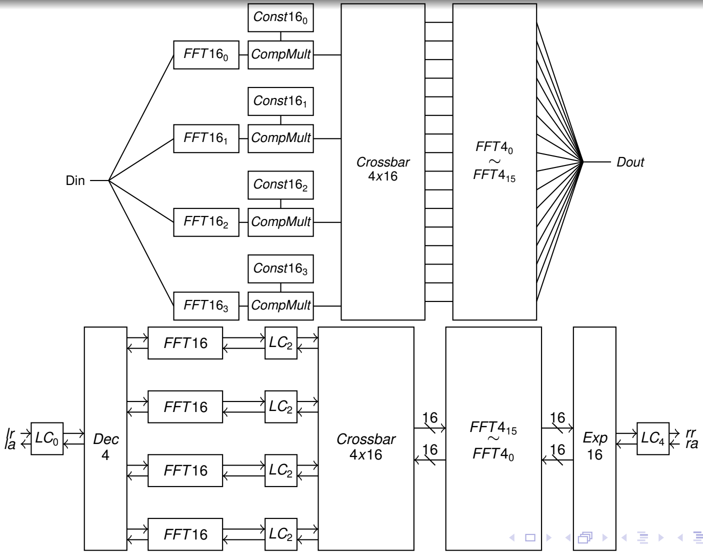
64-Point FFT

- Multi-frequency design with 4 separate frequency domains.
- Hierarchical design methodology, which enables schematic type design development.
- The FFT circuit operates at 1.25GHz and consumes 59.2pJ of energy per data point.
- 2.4×, 2.4× and 3.2× benefit in terms of area, energy and throughput respectively over synchronous design.

General Multi-rate FFT Architecture



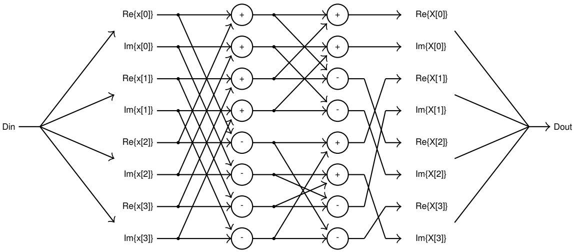
64-point FFT Block Diagram

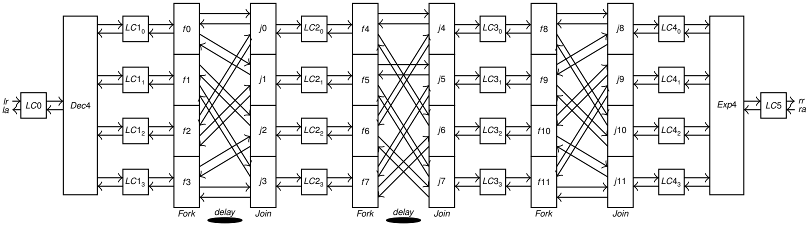


Code Snippet

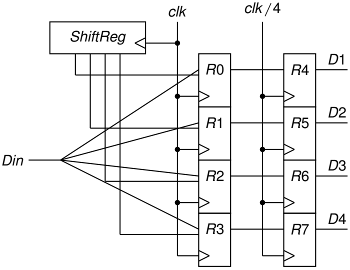
```
module FFT_64 (ri, ai, DI, ro, ao, DO, rst);
input  [`WORD_SIZE-1:0] DI; ...
// input pipeline
linear_control  LC0 (.lr(ri), .la(ai), .rr(p0r),
.ra(p0a), .ck(ck0), .rst(rst));
latch   P0   (.d(DI), .clk(ck0), .q(P0D0));

decimator_4 D4_0 (.DI(P0D0), .D1(P0DT1), .D2(P0DT2),
.D3(P0DT3), .D4(P0DT4),
.ri(p0r), .ai(p0a), .rst(rst),
.r1(p0rt1), .r2(p0rt2), .r3(p0rt3), .r4(p0rt4),
.a1(p0at1), .a2(p0at2), .a3(p0at3), .a4(p0at4));

// The FFT_16 modules.
FFT_16 F16_0 (.ri(p0rt1), .ai(p0at1), .ro(p1rt1),
.ao(p1at1), .DI(P0DT1), .DO(P1DT1), .rst(rst));
```

4-point FFT Block Diagram

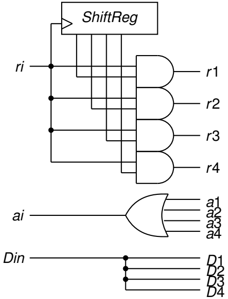




Decimator

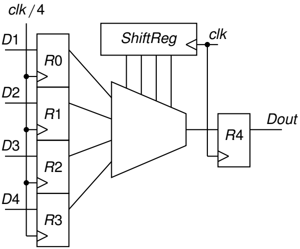


Synchronous Decimator

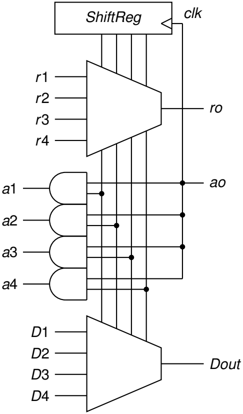


Asynchronous Decimator

Expander



Synchronous Expander



Asynchronous Expander

Results

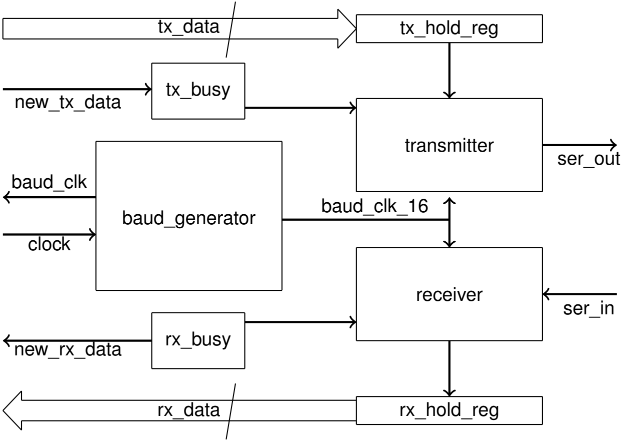
Design Comparisons (+ The nominal process voltage)

Design	$e\tau^2$ Advantage	Normalized FFTs per Energy [3]	Benefit Prod. [1]
This Design (Async-opt)	4,683.38	17.35	68.54
This Design (Async)	5,031.00	18.47	60.37
This Design (Clock)	205.60	7.60	3.23
Baireddy [2]	1.00	–	1.00
Chong (Async-1.1V)	0.02	8.33	4.26
Chong (Async-3.5V)$^+$	0.02	17.01	1.34
Baas (3.3V) [1]	421.98	8.44	14.48
Baas (5V)$^+$[1]	421.98	3.31	9.56

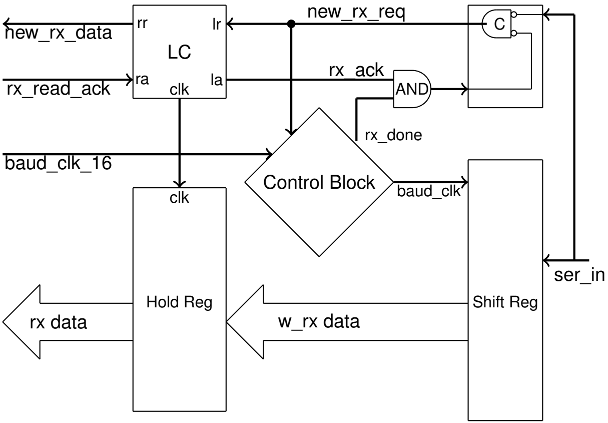
UART

- Frequency dependent slow peripheral.
- There is no way of getting rid of the clock.
- Large idle times which require the receiver line to be polled continuously by a clock signal.
- Reactive nature of asynchronous circuits enables us to identify a valid data reception.
- Asynchronous handshake vs. manual and automatic clock gating.

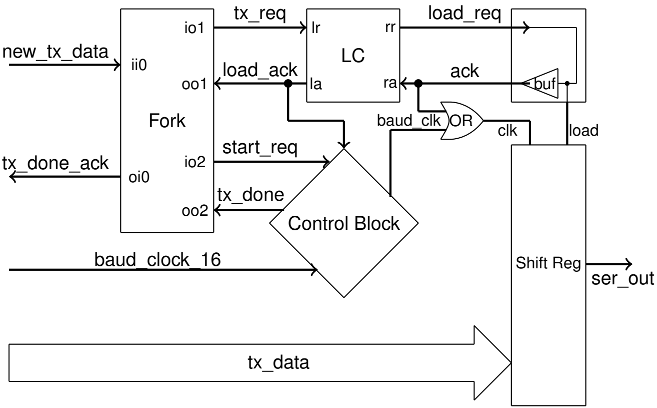
UART Block Diagram



Asynchronous Receiver



Asynchronous Transmitter

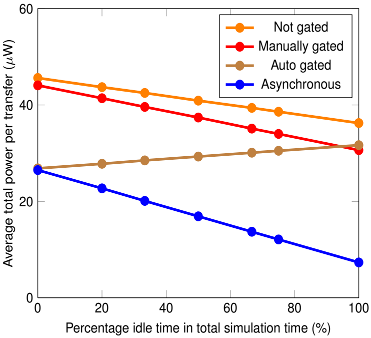


Results

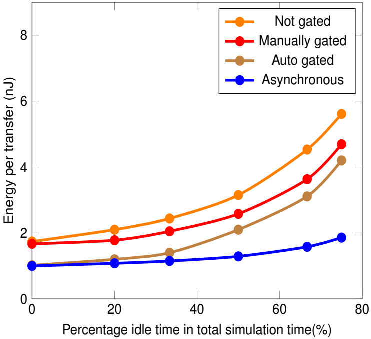
	Area	Power (μW)								Simulation-Time
	Core Area	Active (0% Idle Time)				Idle (100% Idle Time)				
	μm sq	switching	internal	leakage	total	switching	internal	leakage	total	μs
Asynchronous Design	2268.14	4.43	15.38	6.68	26.49	00.00	00.71	06.62	07.33	4000
Clocked – Not gated	2186.94	2.90	37.29	5.42	45.61	2.79	28.18	5.27	36.25	4000
Clocked – Auto gated	2088.94	3.00	18.81	5.04	26.85	3.81	22.88	4.97	31.66	4000
Clocked – Manually gated	2186.75	3.81	34.95	5.32	44.07	2.76	22.68	5.20	30.63	4000

	Area	Power		
	Core Area	Active(100%)	Idle(100%)	Idle(90%)
Asynchronous Design	1.00×	1.00×	1.00×	1.00×
Clocked – Not gated	0.96×	1.72×	5.00×	4.67×
Clocked – Auto gated	0.92×	1.01×	4.32×	3.99×
Clocked – Manually gated	0.964×	1.66×	4.14×	3.89×

Results



Average Power Consumption



Energy per transmission/reception

Conclusion

- Presented a relative timing methodology for developing asynchronous circuits using synchronous CAD tools.
- Design details and choices to generate power benefits in a frequency dependent UART.
- Different approaches are required for different design types.
- Benefits achieved –
  - 64-point FFT - 3.2×, 2.4× and 2.4× benefit in terms of throughput, energy and area respectively over synchronous design.
  - UART - Around 4× power benefit as compared to synchronous design at 10% activity.

Thank You

B.M. Baas.
A low-power, high-performance, 1024-point fft processor.
Solid-State Circuits, IEEE Journal of, 34(3):380–387, 1999.

V. Baireddy, H. Khasnis, and R. Mundhada.
A 64-4096 point FFT/IFFT/Windowing Processor for Multi Standard ADSL/VDSL Applications.
In *International Symposium on Signals, Systems and Electronics*, pages 403–405. IEEE, 2007.

K.S. Chong, B.H. Gwee, and J.S. Chang.
Energy-efficient synchronous-logic and asynchronous-logic FFT/IFFT processors.
IEEE Journal of Solid-State Circuits, 42(9):2034–2045, 2007.

Mito Litochevski.
Uart to bus core specification.
Technical report, Opencores, 2012.

Open Core Protocol.
http://www.ocpip.org/.