Obtaining asynchronous benefits from synchronous design flow

Charlie Brej

cb@cs.man.ac.uk

Amulet Group
Manchester University

Aims

Simple generation of asynchronous designs
(Engineers, Bosses and Marketing)

Use of synchronous tools and experience
(Preserve the production line structure)

Compete with synchronous implementations
(Speed, Power, EMC...)

Transparent synthesis
(WYSIWYG)

Direct translation

Synchronous to asynchronous conversion
(Simple and robust)

Transparent operation
(Full control over the design)

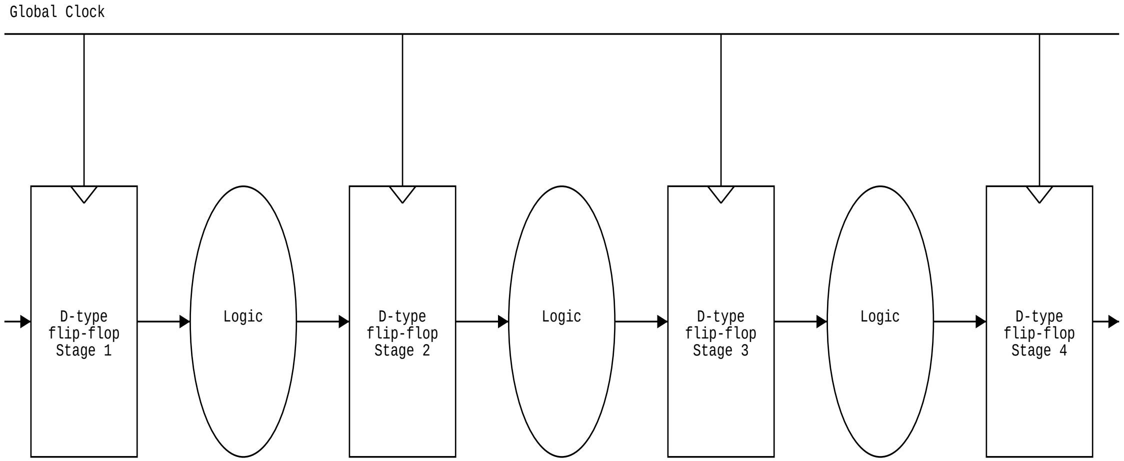
Intrinsically fast conversion
(Desktop computer, quick turn-around time and scalable)

Good input designs give good result designs*
(*usually)

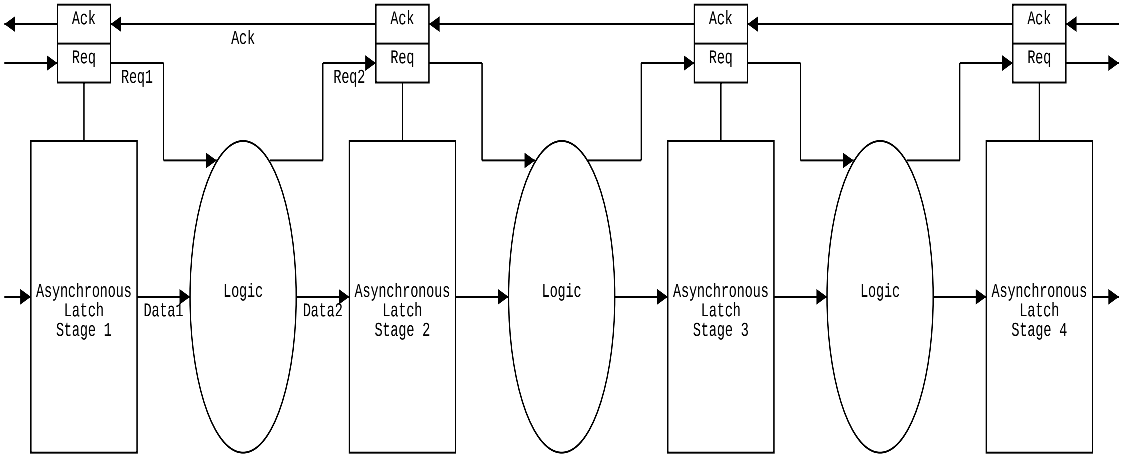
Asynchronous designs

Synchronous Designs

Global clock separates tokens



‘Think Tokens’


Ack
Req
Ack
Req1
Req2
Asynchronous Latch Stage 1
Data1
Logic
Data2
Asynchronous Latch Stage 2
Logic
Asynchronous Latch Stage 3
Logic
Asynchronous Latch Stage 4

Asynchronous Designs

Handshaking separates tokens

Translation

What does the clock do?

1) Stops tokens over-writing each-other

2) Ensures a computation a worst case amount of time

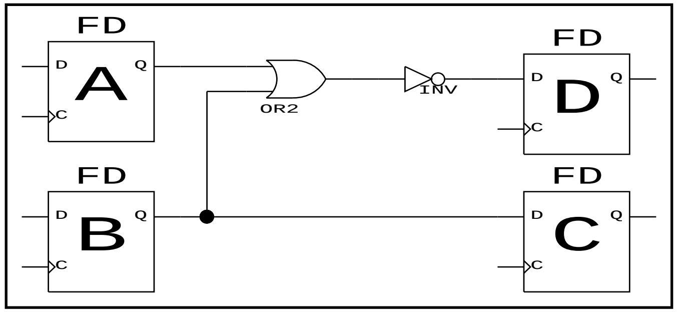
3) Synchronises

Replace clocks with handshakes?

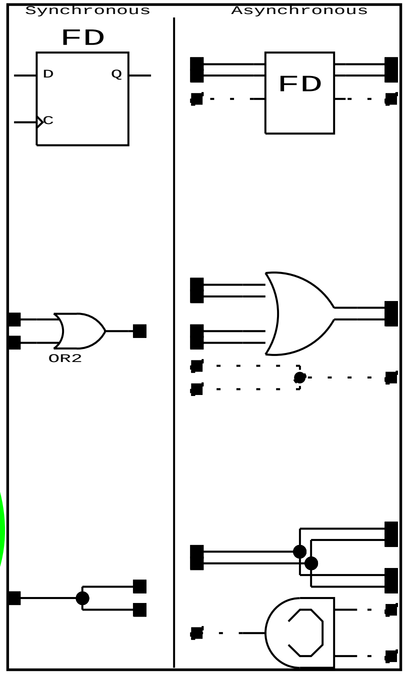
When is the operation complete?

Direct Translation Method

Synchronous

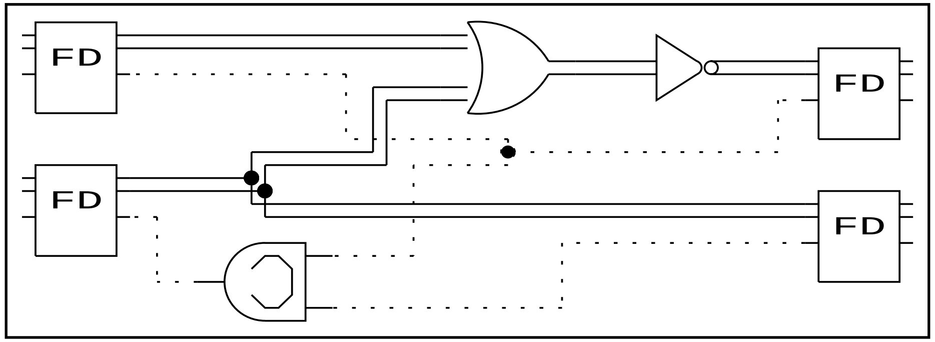


Conversion Table



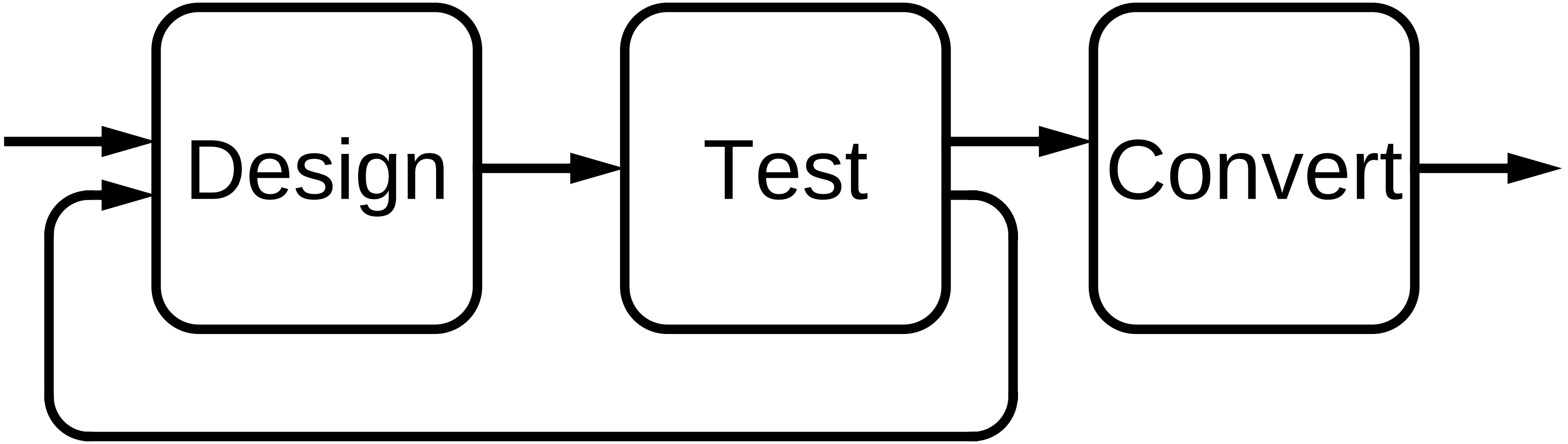
Each element in the synchronous design is replaced using the conversion table

Asynchronous



Conversion table holds replacements for: latches, gates and forks

Tool Flow



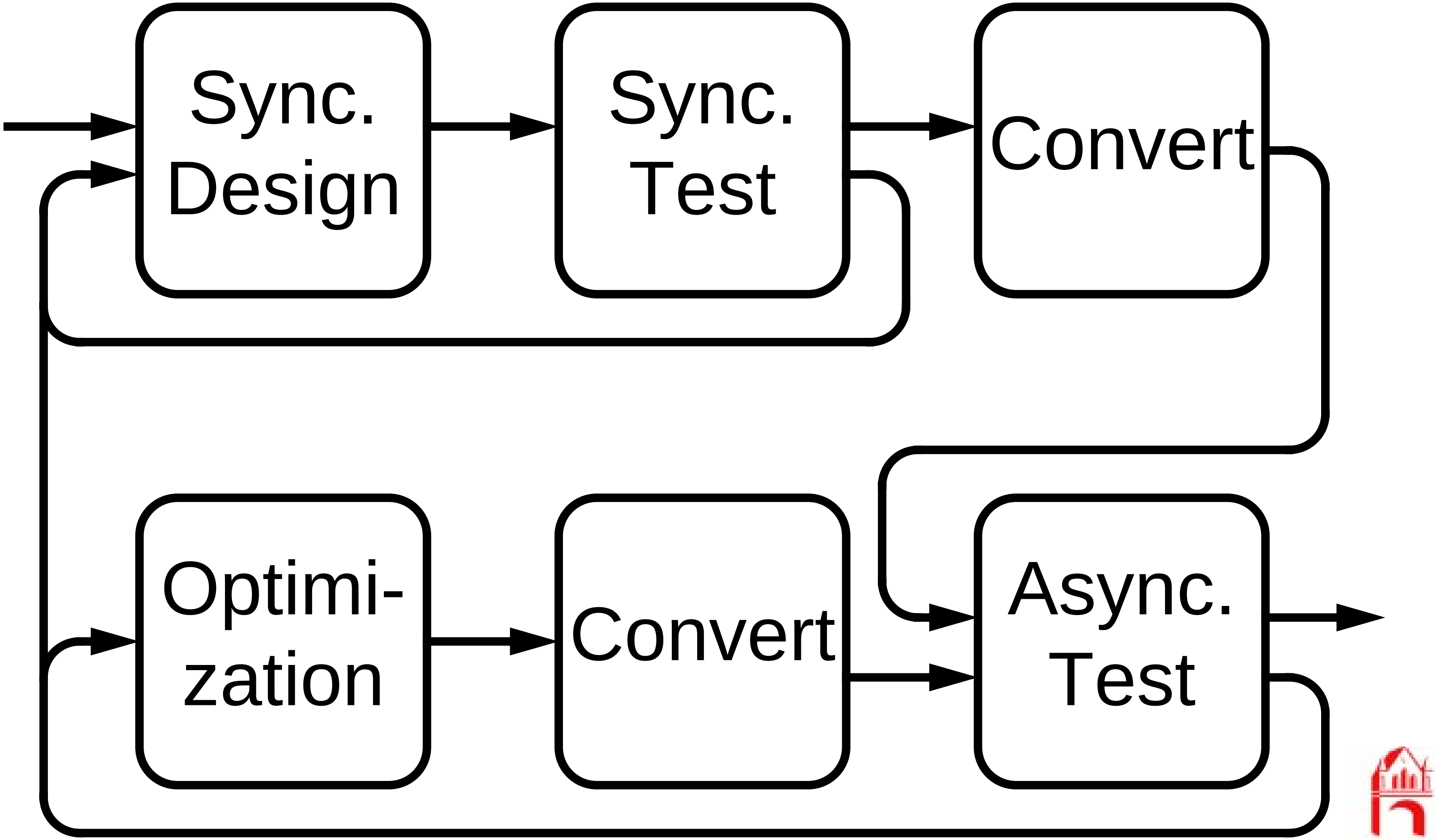
The conversion is kept outside the synchronous design-test cycle

Optimization

Pipelining latches	Increase Pipelining
Semi/Fully decoupled latches	Reduce synchronisation
Anti-token latches	Remove unneeded ‘in-flight’ tokens
1-of-4 data encoding	Reduce power consumption
Conditional read/write elements	Reduce power consumption and synchronization

Retrofit to synchronous source
All elements are ‘safe’ in the synchronous design

Optimization Flow



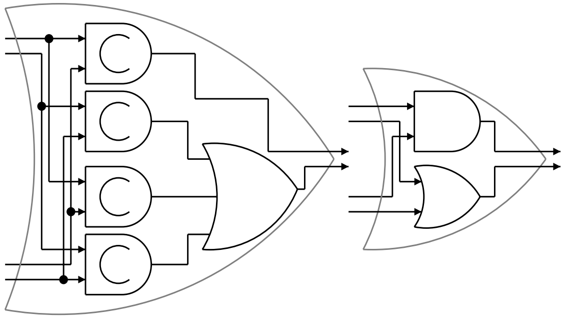
Technology Mappings

DIMS
(Large and Slow but Safe QDI)

Early Output
(Fast, Smaller, Not safe, Fault testing)

Safe Early Output
(Medium speed and size, QDI)

Mixture
(Easy connections between technologies)



DIMS and early output gates

Possible design properties

Bit-level pipelining

Average case performance

Smoothed power signature
(security and EMI)

Faster than synchronous performance
(on some Early Output circuits e.g. Red Star)

Simple super-pipelining
(just add pipelining latches)

THE UNIVERSITY of MANCHESTER

Future work

Optimization element effect and placement rules

Additional optimization elements

System simulation and detection of rules

Automatic optimization element insertion
(placed or hinted)

Automatic simulate-optimize cycle

Summary

Direct translation

Technology mappings

Optimization elements

Good resultant designs

Scalable

Thank you.