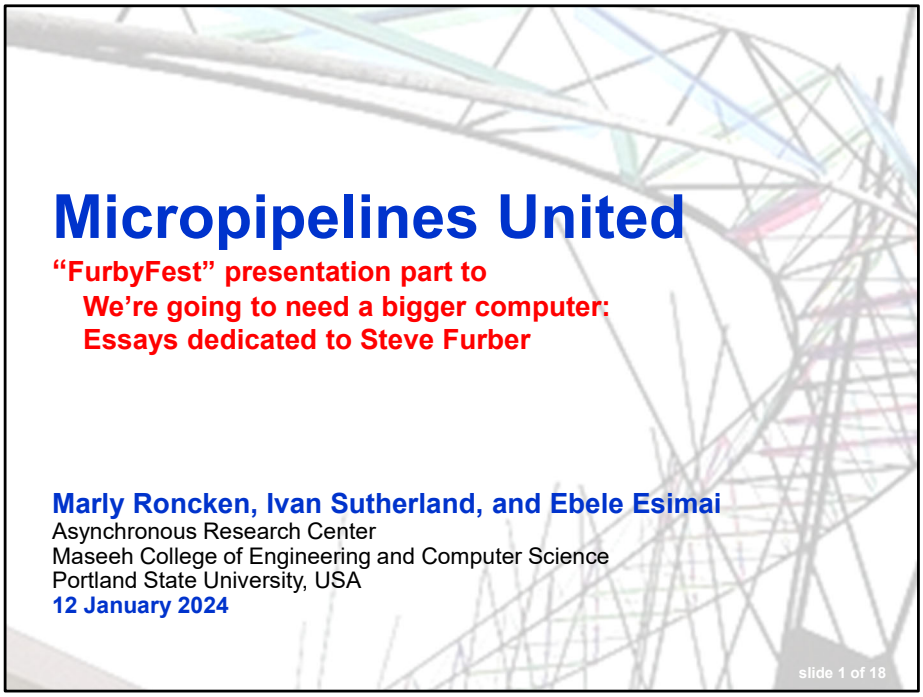
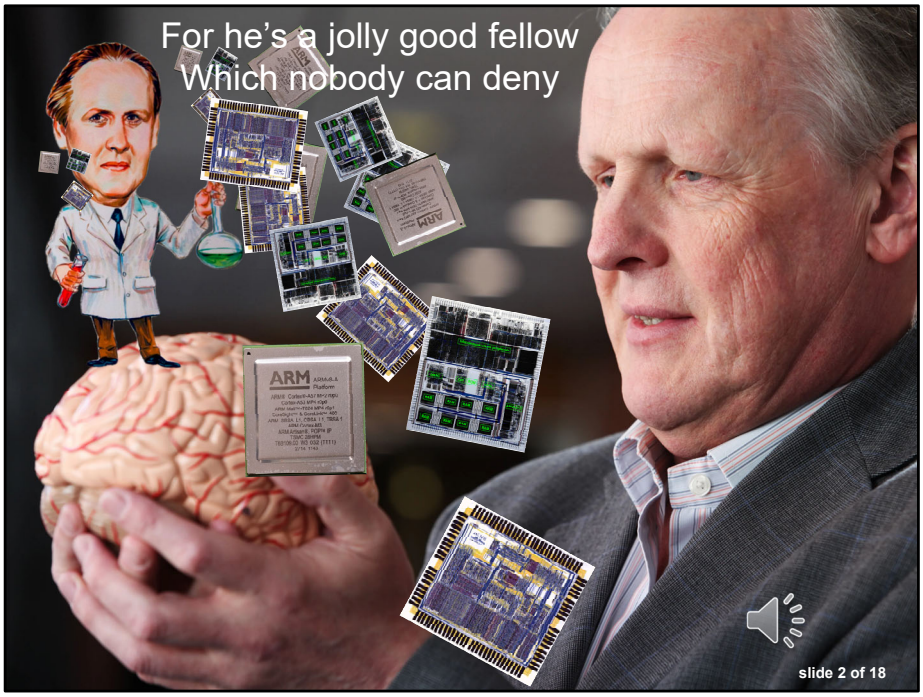


Good afternoon – we are delighted to be here.

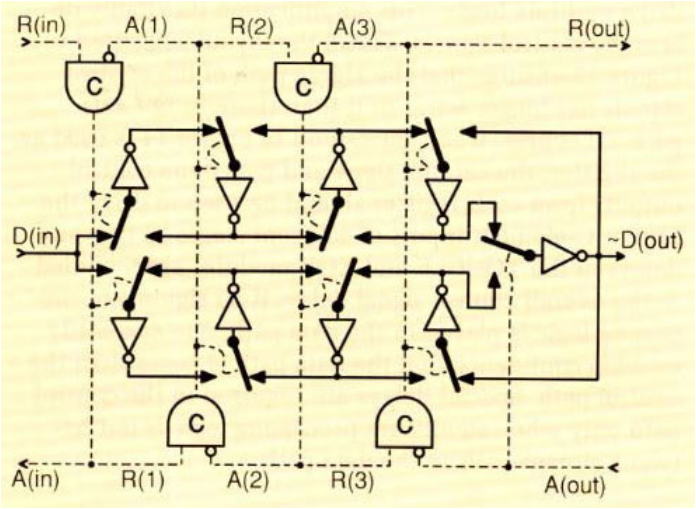
This talk has the title "Micropipelines United"
and is a joint effort by Ivan, me, and our PhD student Ebele.

In this talk, we discuss early asynchronous work
by Steve Furber and the research group at Manchester University
—
work that we have picked up again at Portland State University
because even this early work still reveals new, unexpected
and intriguing properties.


For he’s a jolly good fellow
Which nobody can deny
ARM
slide 2 of 18

History : 90-mile Beach (AU)

I.E. Sutherland, Micropipelines (Turing Award lecture), *CACM* ***32***, 720-738, 1989



FurbyFest 2024 — Micropipelines United slide 3 of 18

I have selected a few historical anecdotes that might be interesting, but before I start I must disavow any attempt to change the letters after Steve Furber's name to FRS(jg) lest someone mistake that for "junior grade." Steve is in every sense of the senior grade.

I learned about the circuit that Charlie Molnar called the "chain of rendezvous" when Charlie was at Washington University in St Louis. In CMOS the chain of rendezvous is built with Muller C-elements like those seen here. Molnar was meticulous about distinguishing

- the logic operation "rendezvous"
- from its level-logic implementation, the Muller C-element.

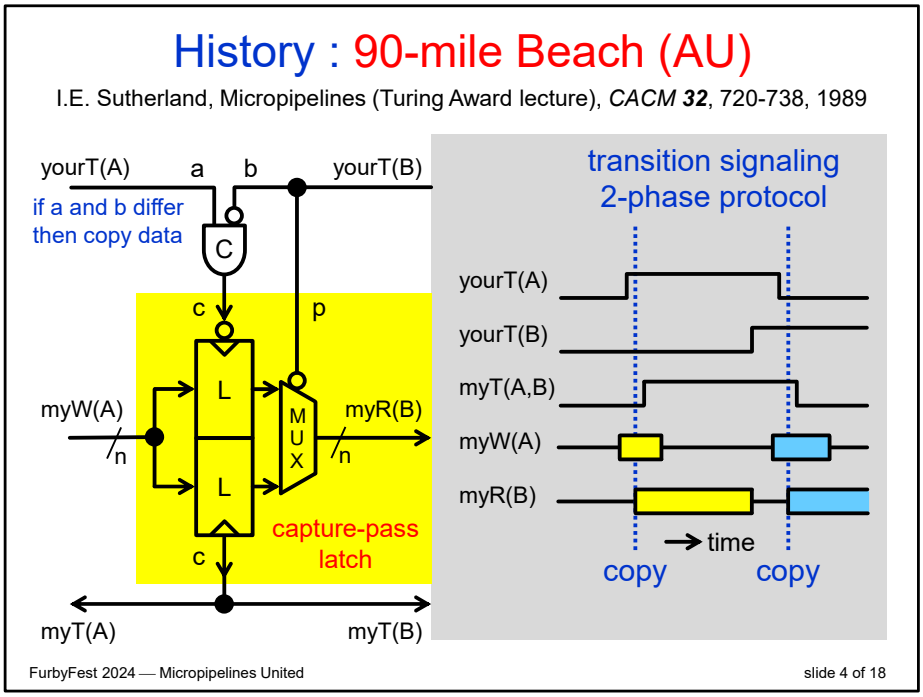
Without level logic one cannot build a Muller C-element although one can build a rendezvous implementation. But please, call it a "rendezvous" or some other name but do avoid calling it a "Muller C-element."

In the late 70s as I was learning about CMOS from Carver Mead at Caltech I showed Claude Shannon a pass-transistor circuit of which I was proud. Claude promptly showed me a decades-old relay circuit for the same function. Shannon was, after all, an expert in relay switching circuits, having applied Boolean Logic to switching circuits for the first time as his 1937 (before I was born) Master's thesis at MIT. Representing pass transistors as switches often helps understanding, as in this Figure.

Notice please the symmetry of this Figure. It has

- up/down symmetry,
- left/right symmetry
- and true/false symmetry.

After first drawing the circuit, I was sure that it could not possibly work because there was no "input" or "output" distinction to its gates. Without clear input and output, how could the circuit possibly know in which direction to move data? Only after David Douglas built the circuit when I returned to Pittsburgh from Australia did I know that the circuit actually worked.



In 1990 I made a special trip to the University of Illinois to meet David Muller. There I learned that in the early days his group labeled the inputs of his famous circuit A and B and its output C, from which it got its name.

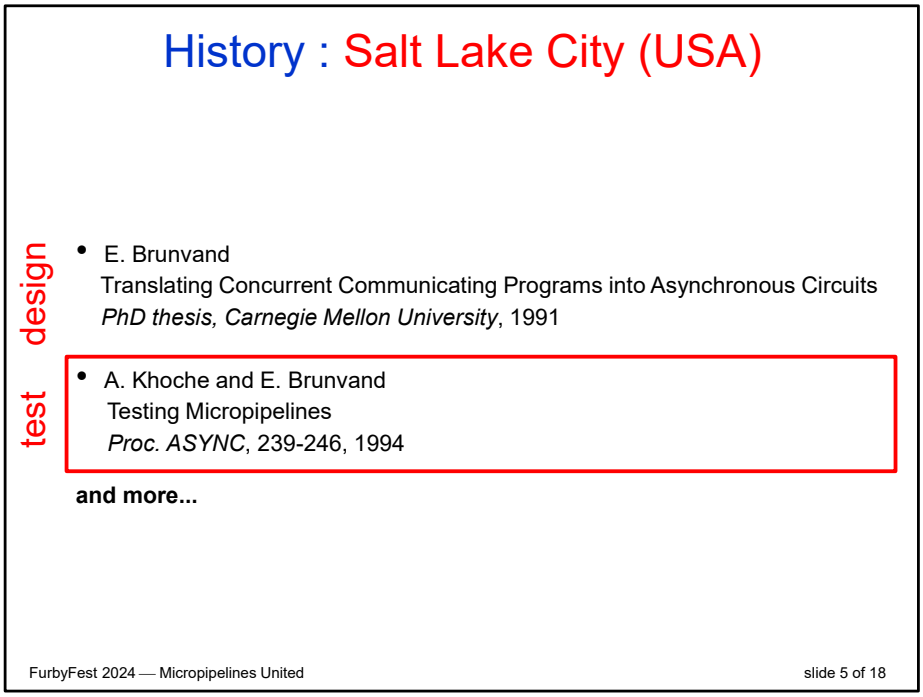
During my talk there I used a row of the audience to demonstrate the chain of rendezvous circuit. Naturally, I picked the row in which Muller himself sat. I asked each person in that row to use a raised or lowered hand to indicate their state. Each person followed the rule:

“If predecessor and successor differ, copy predecessor.”

The flow of raised and lowered hands demonstrates the self-timed action. Although Muller had invented the C-element, he admitted that this was the first time he had ever “been” a Muller C-element.

Sometime in 1987 I got a phone call from the ACM asking “if I would accept a Turing Award!” (Is the Pope a Catholic?). I would be required to offer an award paper for publication in ACM communications. I saw this requirement not as a chore, but as an opportunity for a publication free of peer review! The 1989 Micropipelines paper is essentially a tutorial about transition signaling built around the strange form of First-in-first-out (FIFO) device that I had devised during an extended Australian vacation away from winter snow, on 90-mile Beach.

The Micropipeline’s excessive symmetry makes the Micropipeline circuit prone to interchange errors and therefore hard to make correct.
The Figure on THIS slide and those that follow avoid confusion by using ordinary latch symbols with distinct inputs and ouputs.

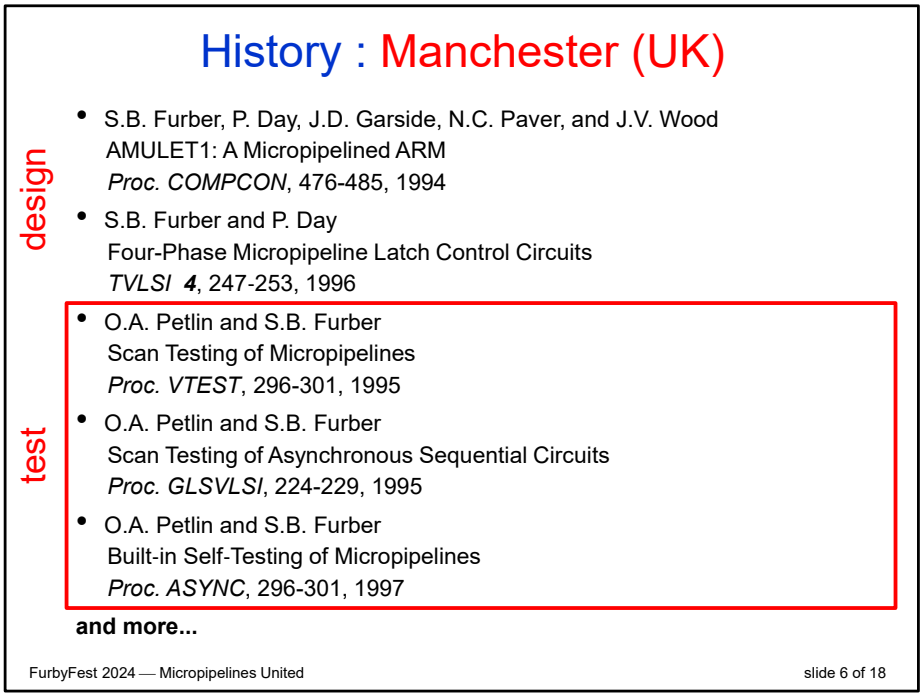


The Micropipeline paper defines an asynchronous circuit family
– a blueprint to design asynchronous circuits.
It may be the first asynchronous family
– I don't know that for sure.

But I do know that the paper spiked
new interest into asynchronous design.

Three locations in particular took up Micropipelines.

The first location is Salt Lake City.
Erik Brunvand, who gave a talk before lunch, worked on
Micropipeline design compilation for his PhD thesis at CMU.
He then moved to the University of Utah,
where he worked with PhD student Ajay Khoche
on testing Micropipelines.



The second location is Manchester.

This is of course the group of Steve Furber
at the University of Manchester, here in the UK.
Steve's group had the ambitious target to build
a testable asynchronous ARM processor.

We mention only a few early publications,
in particular the ones on testing
by Steve and his PhD student Oleg Petlin.

These early publications started with
- Micropipelines with capture-pass latches
- as presented by Ivan

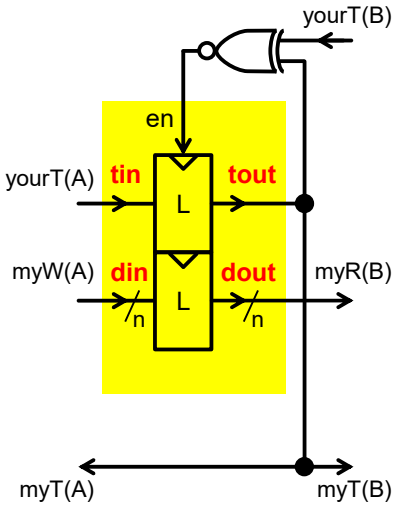
but quickly moved to
- Micropipelines with ordinary latches
- that store data only half the time in half the area.

Today, we know that there is a better way to do
Micropipelines with ordinary latches.

History : New York City (USA)

M. Singh and S.M. Nowick, MOUSETRAP: High-Speed Transition-Signaling Asynchronous Pipelines, *TVLSI* ***15***, 684-698, 2007

- different design
- same interface
- same protocol



FurbyFest 2024 — Micropipelines United

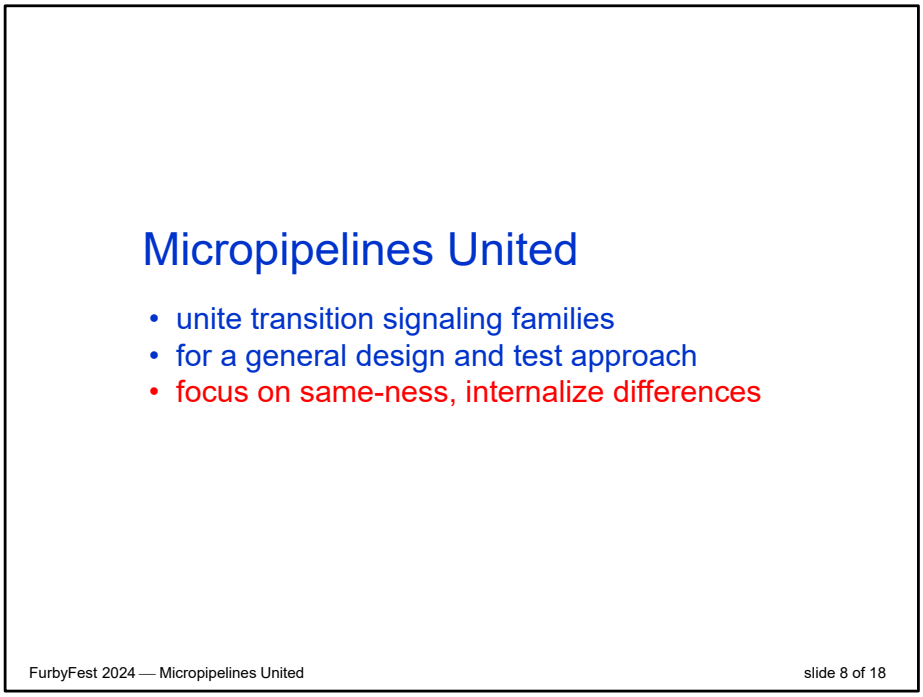
slide 7 of 18

But this better way wasn't invented until 10 years later by Montek Singh and Steve Nowick, in New York City.

It is called "Mousetrap."

Mousetrap uses ordinary latches.
but it has
- the same interface
- and the same transition level protocol

as a Micropipeline.

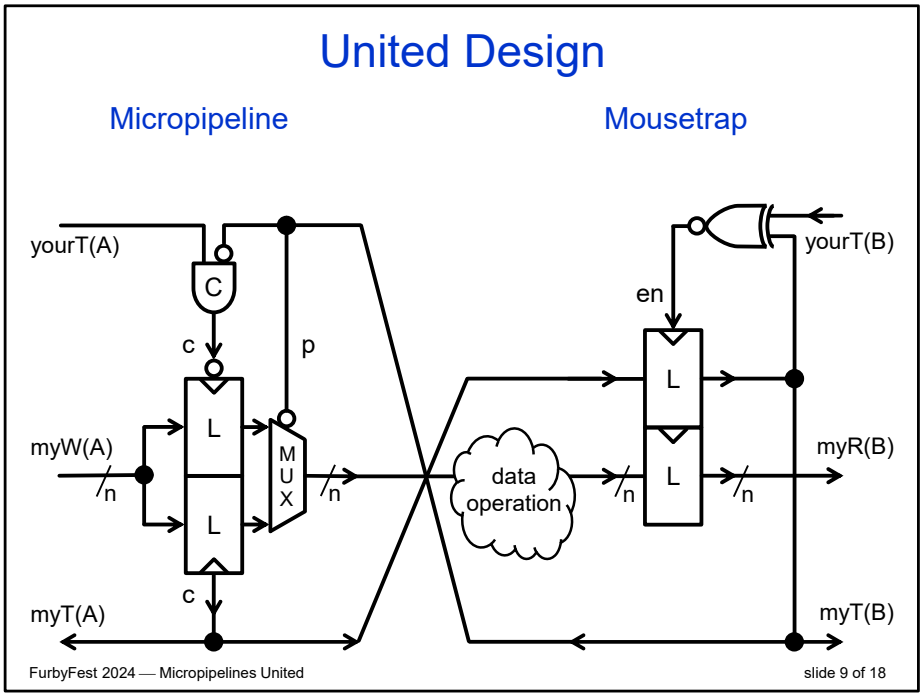


Because Micropipeline and Mousetrap use the same protocol,
we can unite the two into
- a single design and test approach
- for transition signaling.

But we want to do more than that.

We want to identify
- parts that are the same between the two families
- from parts that are different

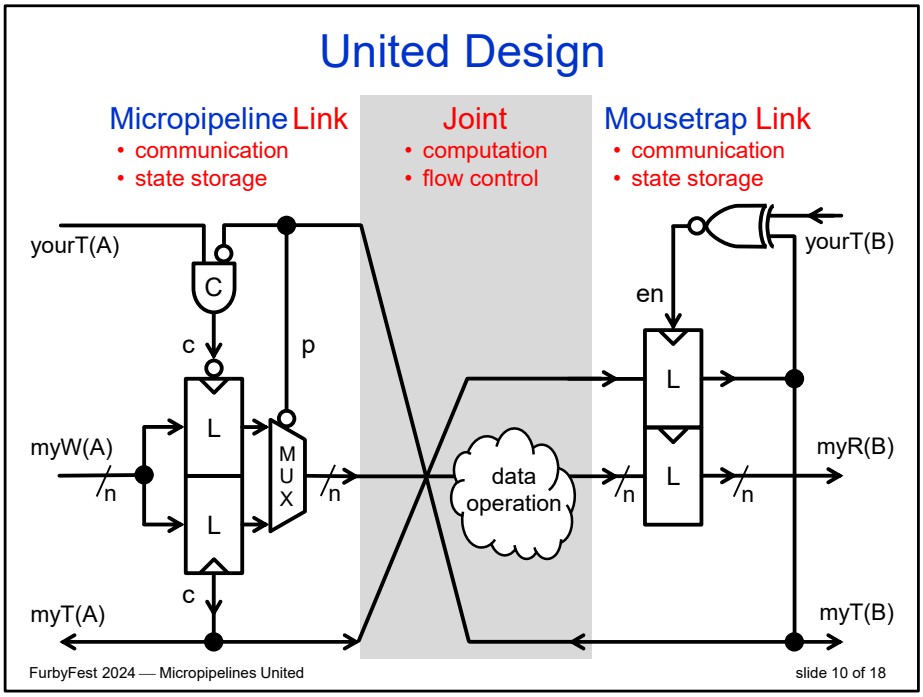
so we can make a more general design and test approach.



Let’s first look at a united design approach.

We have a Micropipeline design on the left
and a Mousetrap design on the right.

They connect as follows:
<CLICK>
Any data operation between them goes here:
<CLICK>



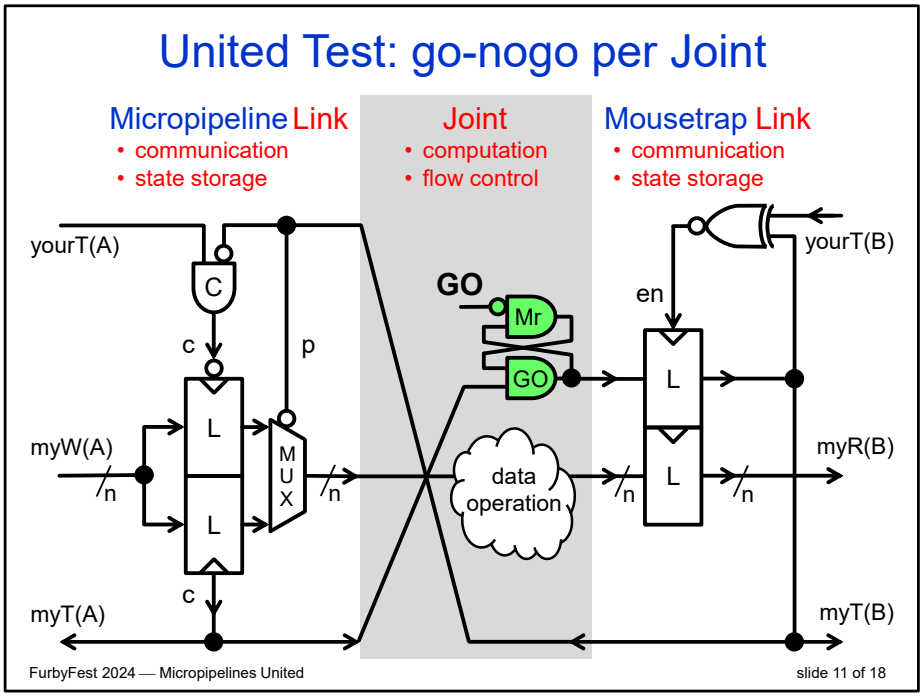
Because the middle part is shared
and is the same for Micropipeline and Mousetrap,
we turn the middle into a separate module called “Joint.”
<CLICK>

The remaining Micropipeline and Mousetrap parts differ only internally
We turn each into a separate module called “Link.”
<CLICK>

This partitions the design into
- two Links that store state

and
- one STATE-LESS Joint that acts on Link state
- and controls how the Links exchange state information.

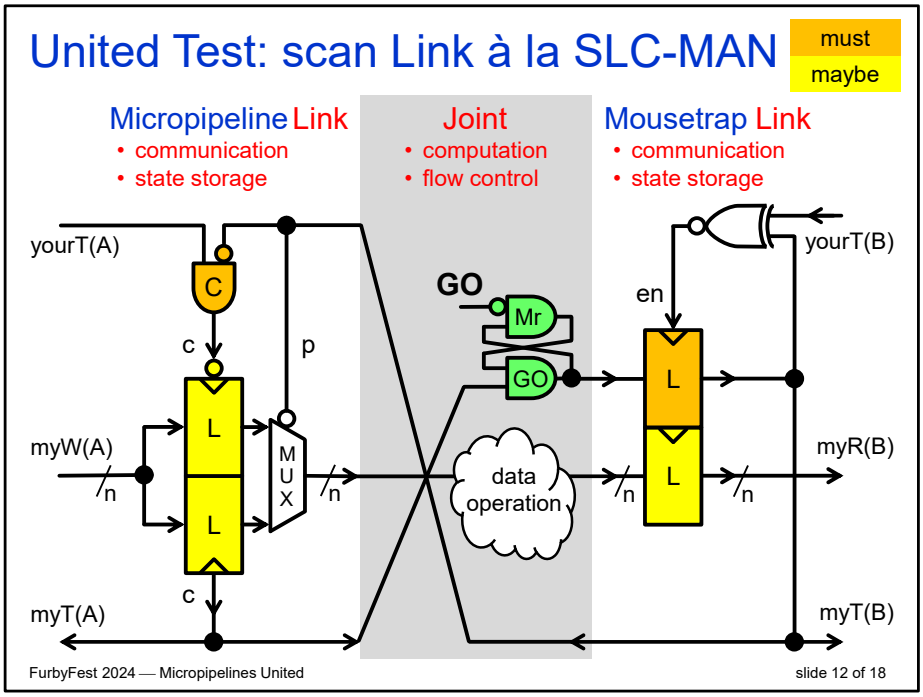
We use similar Link-Joint partitions
for circuit families with level-signaling protocols.
The fact that this partition works for multiple signaling protocols
is a good sign that Links and Joints
provide a generic approach to design and
- as we shall see– to test as well.



Now, let's look at a united test approach.

To enable and disable self-timed actions
- we add a go-nogo signal and gate to each Joint.

Our level-signaling Link-Joint designs use a similar go-nogo solution. The gate is called "MrGO."



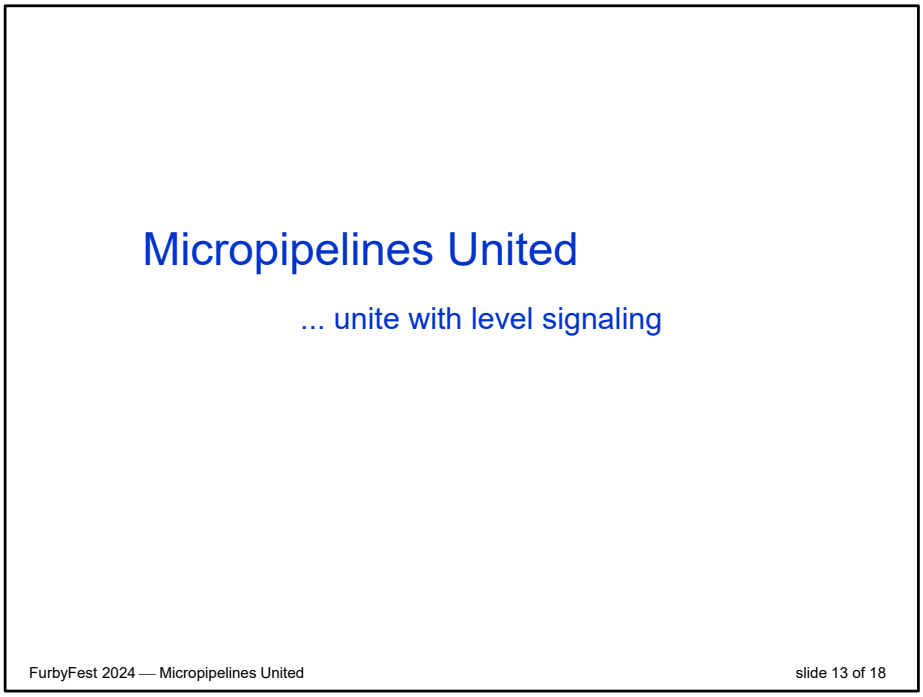
When all Joint actions are disabled,
it is safe to examine and initialize the Link states.

We do this using scan test.
Our scan solution
- combines the scan test solutions
- from Salt Lake City and Manchester.

We scan all orange-colored states
- these are the states that control the protocol

We may or may not scan the yellow-colored states
- these are the states that store the data

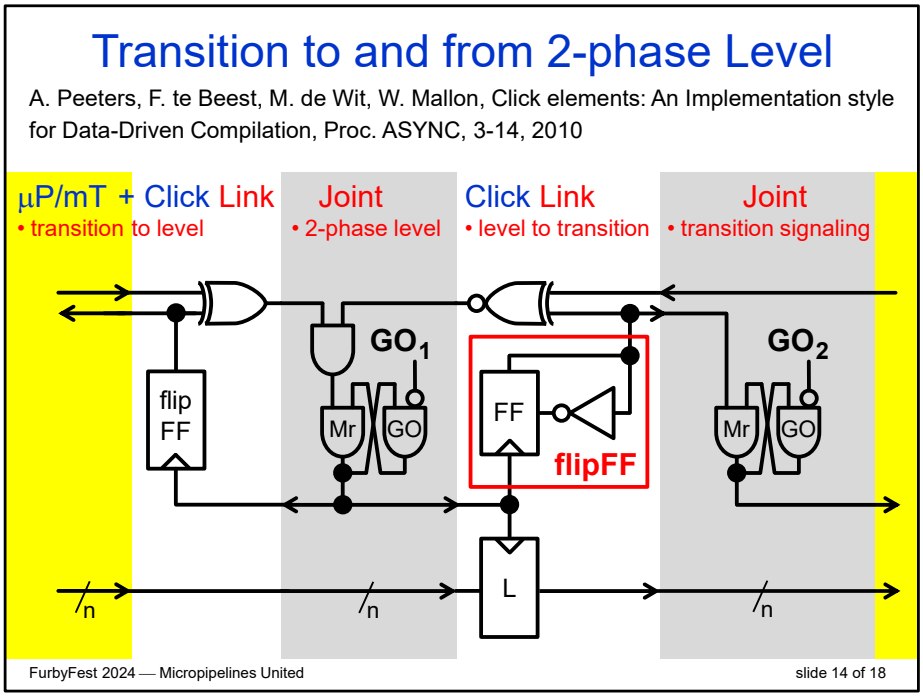
When the old Link state have been examined
and the new Link states have been scanned in,
- we enable the Joints
- to run the circuit.



So far, our examples show only
- united design and test solutions
- for transition signaling
- with basic operations.

More complex operations tend to require level signals.

So, to do both basic and complex operations, it makes sense to unite
- transition signaling with level signaling.



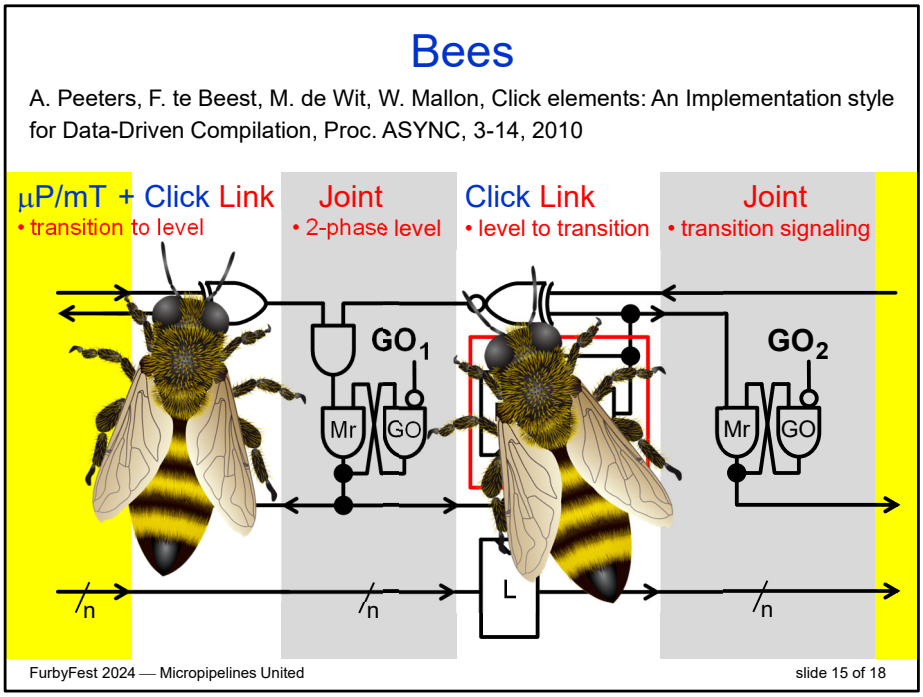
This slide shows how to unite
- transition signaling
- with 2-phase level signaling

The yellow-colored parts on the left and right
- contain transition signaling circuits
- implemented in Micropipeline or Mousetrap

The two grey-colored parts contain Joints.
The left Joint uses 2-phase level signaling.
The right Joint uses transition signaling.

We unite these parts with two Links
- one for transition to level signaling
- and another for level to transition signaling

<CLICK>
The two Links use a circuit family called Click.



The best part of this solution is what you DONOT see.

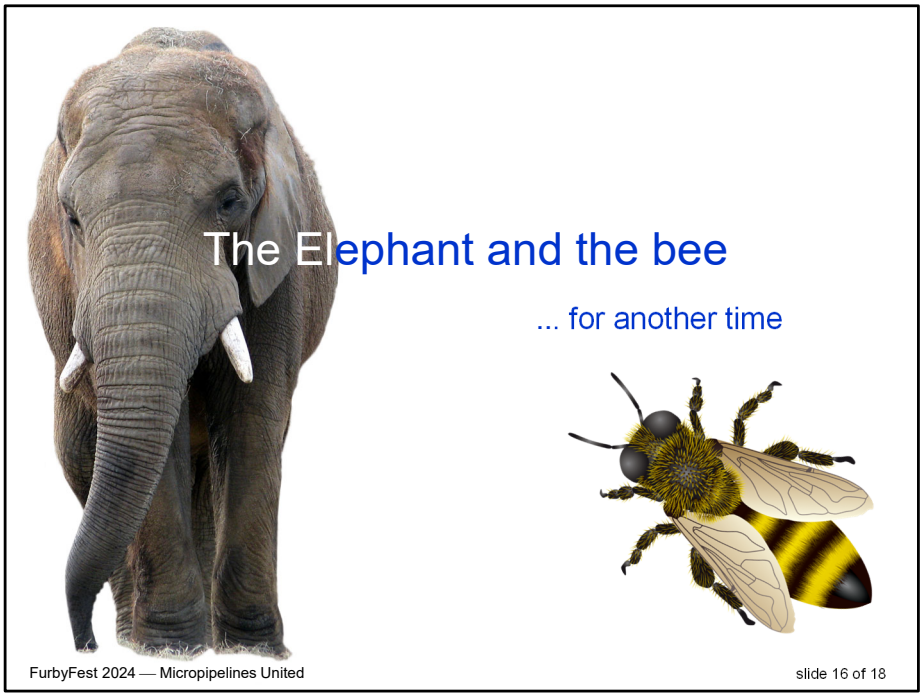
If this were a ring of modules, then you DONOT see
the ring biting its own tail and choking.

You DONOT see this because the Click flipping flipflops
keep the left and right transitions in the correct phase,
and separated so the left (tail) and right (head) don't bite each other.

We call this bad thing that may happen if you use transition signaling
"the elephant in the room."

We call the Click flipping flipflops "bees"
<CLICK>

Bees keep elephants away, because elephants fear bees.



There are a number of youtube videos
- about fences with beehives
- that prevent elephants eating up crops.

You can buy the bees' honey as elephant-friendly honey.
It is elephant-friendly because the bees avoid the need to kill the elephant!

For transition signaling, the elephant and the bee
lead to very interesting new design explorations.

But that's for another time.

SPECIAL THANKS

Manchester (UK)
- Oleg Petlin
- Steve Furber

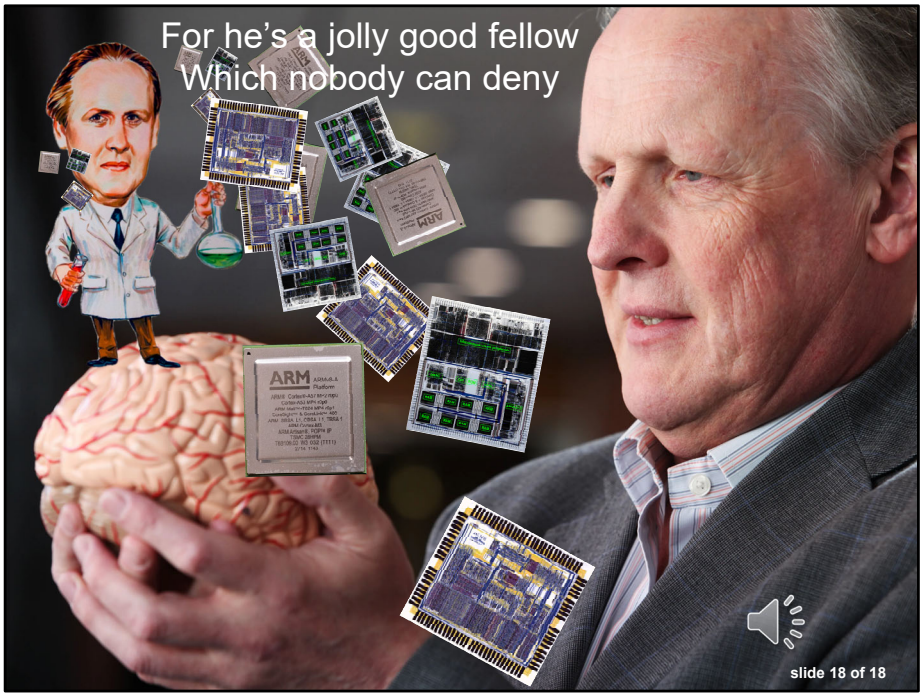
Salt Lake City (USA)
- Ajay Khoche
- Erik Brunvand

New York City (USA)
- Montek Singh
- Steve Nowick

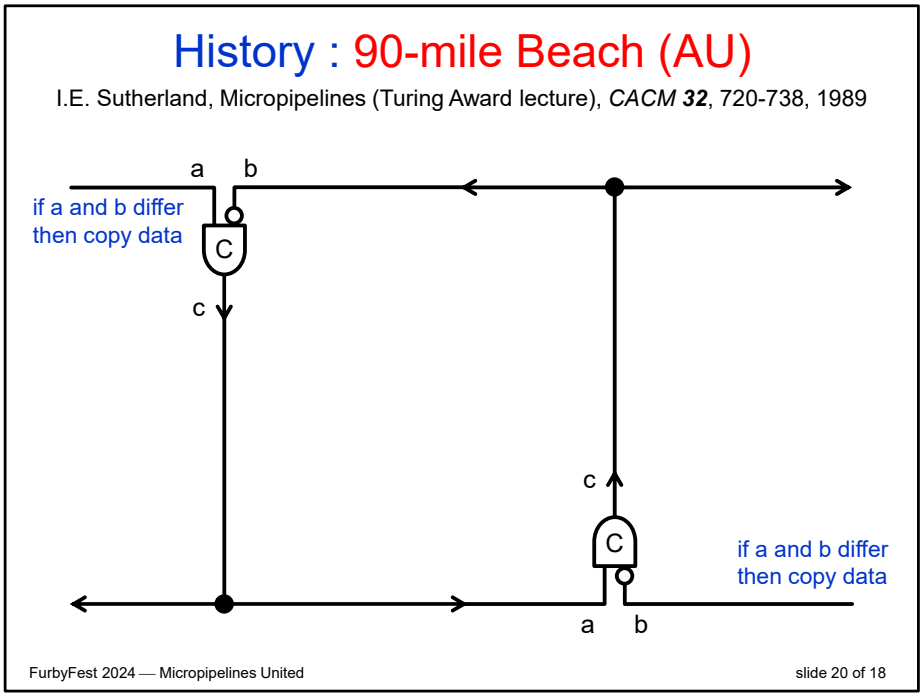
Eindhoven (NL)
- Ad Peeters
- Frank te Beest
- Mark de Wit
- Willem Mallon

SPECIAL THANK YOUs
for keeping our research life interesting
go to:

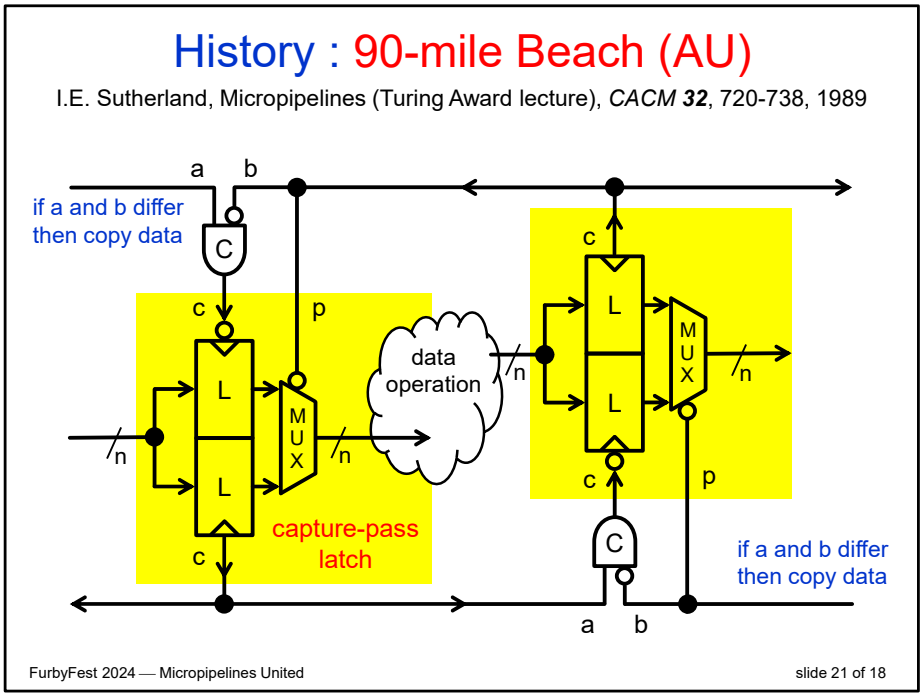
Oleg, Steve,
Ajay, Erik,
Montek, Steve,
Ad, Frank, Mark, and Willem.


For he’s a jolly good fellow
Which nobody can deny
ARM
slide 18 of 18

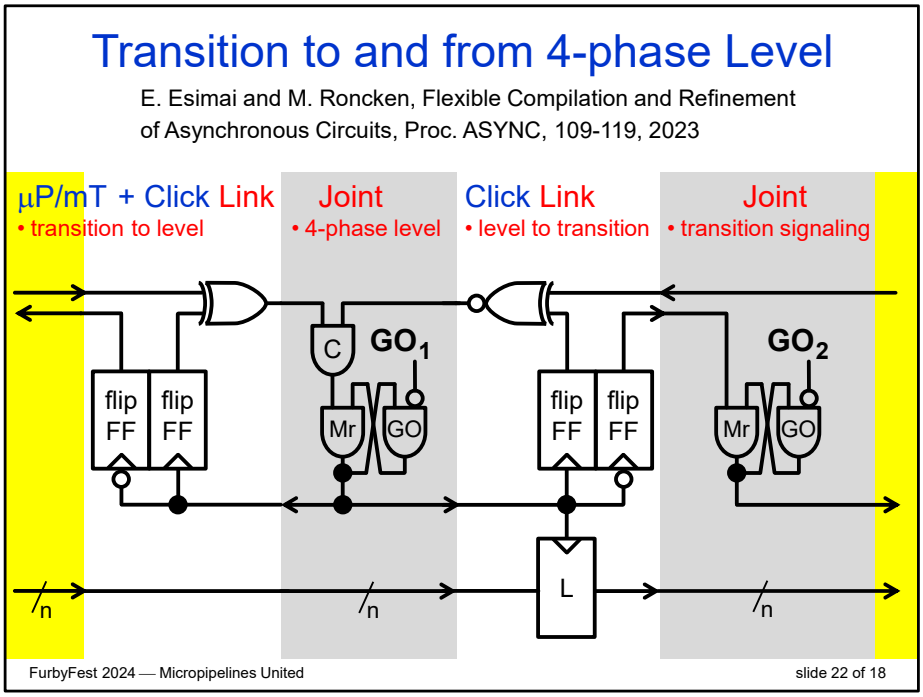
Appendix



The C element “chain of rendez-vous
forms the data-less core of Micropipelines,
and often serves as reference for the logo
in each international (and mostly yearly) ASYNC conference.



Connecting two Micropipeline stages.


Transition to and from 4-phase Level
E. Esimai and M. Roncken, Flexible Compilation and Refinement
of Asynchronous Circuits, Proc. ASYNC, 109-119, 2023
μP/mT + Click Link
• transition to level
Joint
• 4-phase level
Click Link
• level to transition
Joint
• transition signaling
flip FF
flip FF
C
GO1
Mr
GO
flip FF
flip FF
GO2
Mr
GO
L
n
n
n
FurbyFest 2024 — Micropipelines United
slide 22 of 18

(see paper)