Asynchronous Computing

1. Introduction

Marly Roncken and Ivan Sutherland
Asynchronous Research Center (ARC)
Maseeh College of Engineering and Computer Science
Portland State University
July 2018

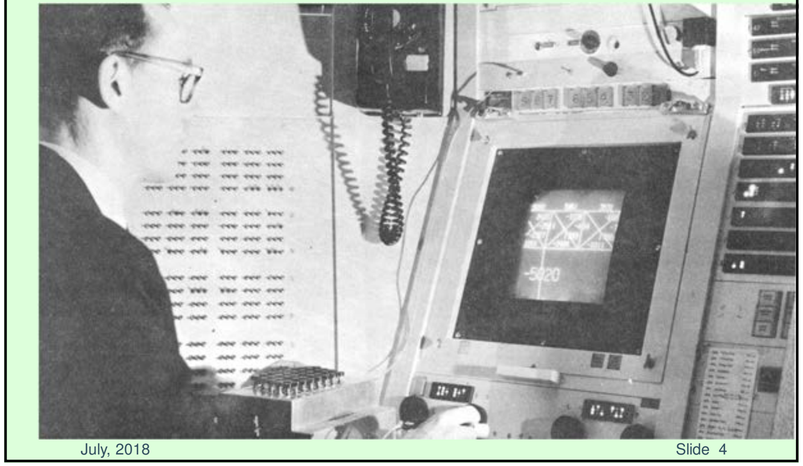
Who we are:

- Marly Roncken
  - Born Netherlands
  - Utrecht, Philips, Intel, ARC
- Ivan Sutherland
  - Born USA 1938
  - MIT, Gov’t, Startup, Caltech, Sun, ARC
- We met at ASYNC 1994
- Married 2006
- 2009 started ARC in Portland, Oregon

Five parts

- Introduction (Ivan)
  - The “clocked design paradigm” used now
  - The “asynchronous” or “self-timed” future
- Link and Joint model (Marly)
- Arbitration (Ivan)
- Initialization, Test & Debug (Marly)
- Discussion + Q&A

Sketchpad (1963)



TX-2 computer (1958 – 1978)

100 KHz clock = every ten microseconds

Light goes 3 km in ten microseconds

Wires very much faster than logic

Can ignore Wire delay

Clocked design paradigm

- Logic acts on each clock tick
- Assume instant data transport to other logic
- Do next step on next clock tick
- All logic marches in step to the clock beat
- Step by step progress
- Very easy to understand

Clocked design paradigm

- **Ignoring data transport delay makes clocked logic very easy to understand**
- So easy that clocked design is now almost universal

Chess: a clocked metaphor

Chess moves are like the steps of clocked logic.
Between moves all pieces are stable.
Each move is instantaneous.

Chess: a clocked metaphor

Question: who runs Faster?
A Horse (knight) or Chariot (rook)?

Chess: a clocked metaphor

Question: who runs Faster?
A Horse (knight) or Chariot (rook)?
The question is meaningless.

Chess players

- Lack any notion of how fast pieces run because all pieces move instantly
- Lack a vocabulary of running speed
- Lack a way to reason about arrival time
- Strategy needs only **where** and not **when**

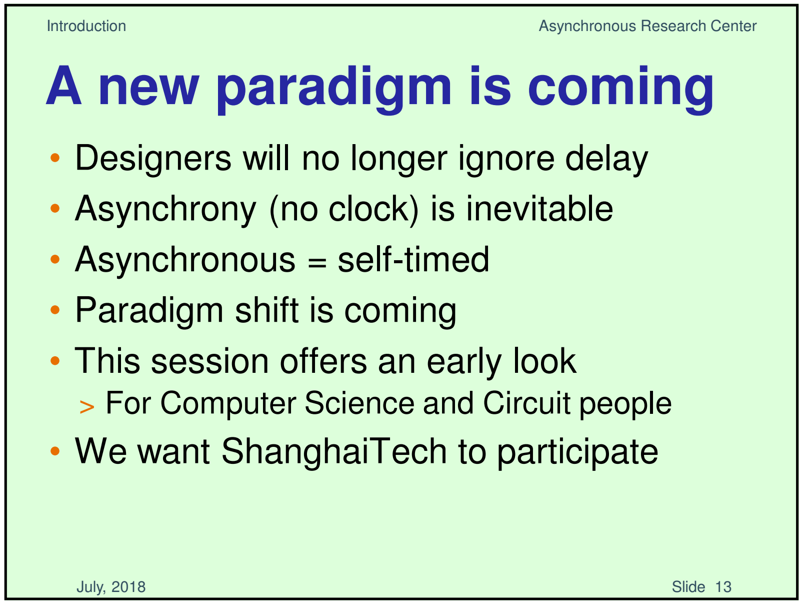
- Asking a chess player which piece is faster is like asking which digit is faster the digit “4” or the digit “7”

Clocked design is failing

- **Ignoring data transport delay makes clocked logic very easy to understand**

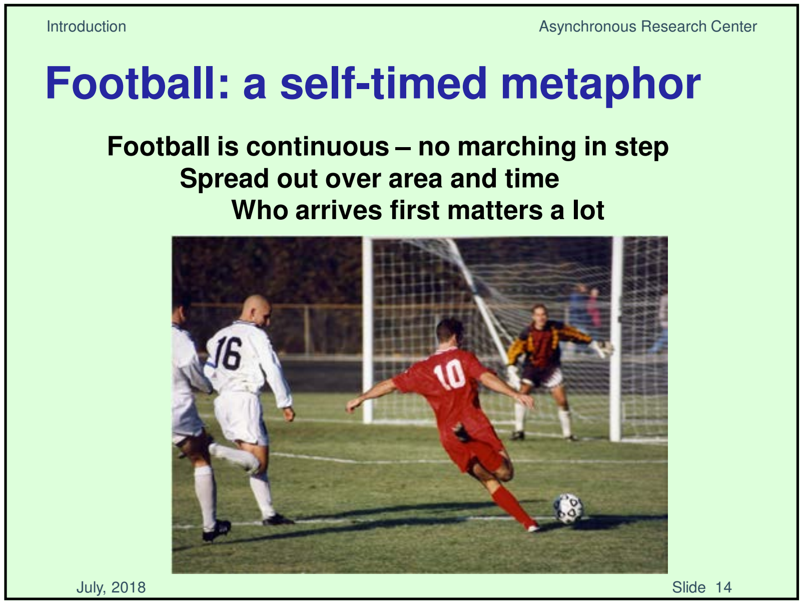
. . . BUT

- **Transistors are now so fast and chips are now so big that data transport delay now matters**



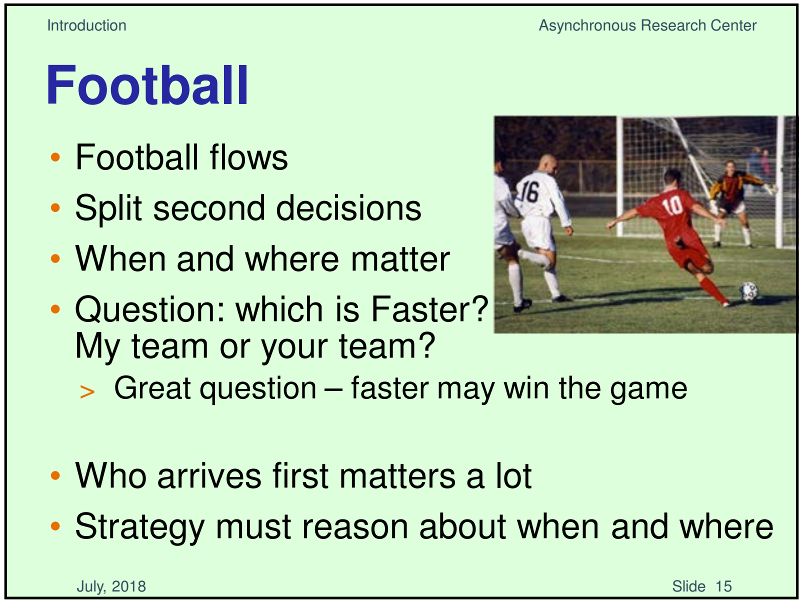
A new paradigm is coming

- Designers will no longer ignore delay
- Asynchrony (no clock) is inevitable
- Asynchronous = self-timed
- Paradigm shift is coming
- This session offers an early look
  - For Computer Science and Circuit people
- We want ShanghaiTech to participate



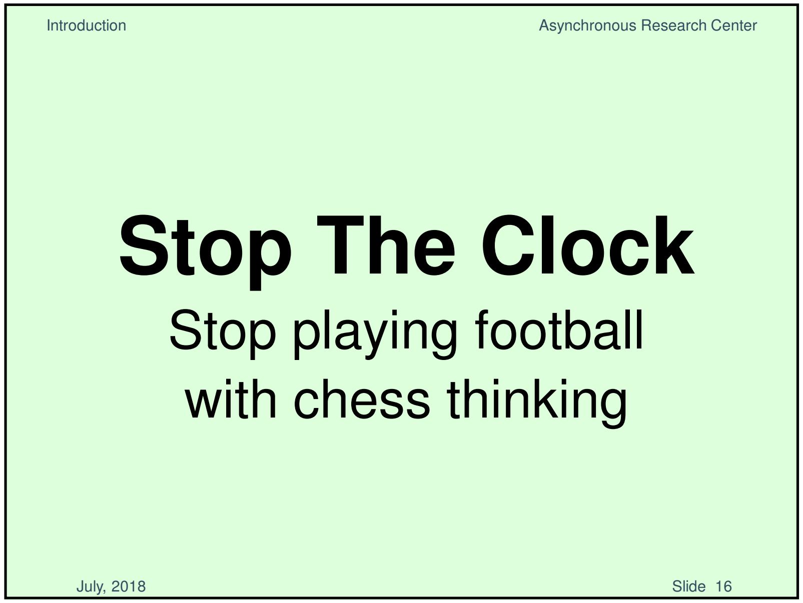
Football: a self-timed metaphor

Football is continuous – no marching in step
Spread out over area and time
Who arrives first matters a lot



Football

- Football flows
- Split second decisions
- When and where matter
- Question: which is Faster? My team or your team?
  - Great question – faster may win the game
- Who arrives first matters a lot
- Strategy must reason about when and where



Stop The Clock

Stop playing football
with chess thinking

We can do better

The self-timed paradigm

- Asynchronous = self-timed data transport
  - Nearby arrives soon; further takes longer
  - Say when data arrive
- Asynchronous = self-timed operations
  - Easy is fast; hard takes longer
  - Say when each action is done
- Use logic gates to schedule
  - When to act
  - When to transport data or fetch new data

Need concurrent thinking

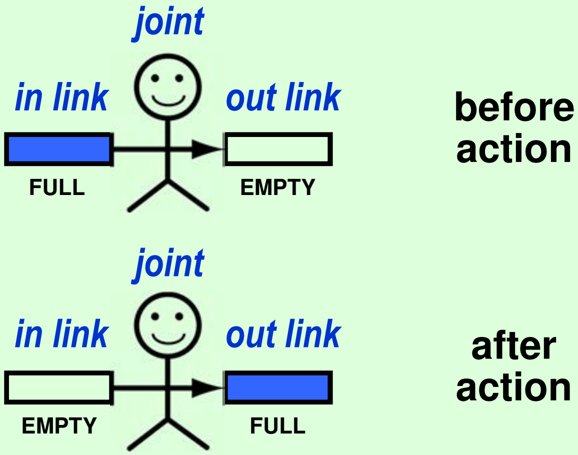
- Hard to think concurrently
- Takes a new point of view
- To understand a beehive
  - It’s not a chessboard
  - Follow one bee at a time
  - Think like a bee
- What does each bee do?

Point of view: self-timed

- Every action reports when it’s done
- Like a software subroutine return
  - Carries an answer AND
  - Allows next code to proceed
- Every data transport reports arrival
- A vocabulary for talking about WHEN
  lets us apply logic to sequencing actions
  rather than marching with everyone in step

Point of view: data

Point of view: action



We call it

The Link and Joint model

Links transport + store data

Joints compute + control flow

Equal partners

The Link and Joint model

- Roncken et al., *"Naturalized Communication and Testing,"* ASYNC-2015
- Roncken et al., *"How to Think about Self-Timed Systems,"* Asilomar Conference on Signals, Systems, and Computers, 2017