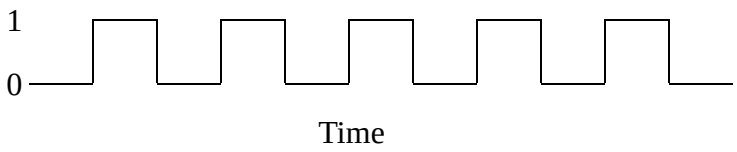
Timing in Asynchronous Circuits

What do we mean by clock?

- The system clock for an integrated circuit is a voltage signal that pulses at a regular frequency.



- The clock tells each stage of a circuit that the inputs of that stage are valid and can be processed.

Current PC Technology

- One of the defining performance characteristics of current personal computers is the clock speed of the microprocessor.
- Current home PC microprocessors approach **3 GHz**.
- The Pentium 4 chip has an area of around 217 mm^2, or around 15 mm from edge to edge.
- A clock signal must travel the 15 mm distance in less than .333 nanoseconds = **100 million miles/ hour**.

What does this mean?

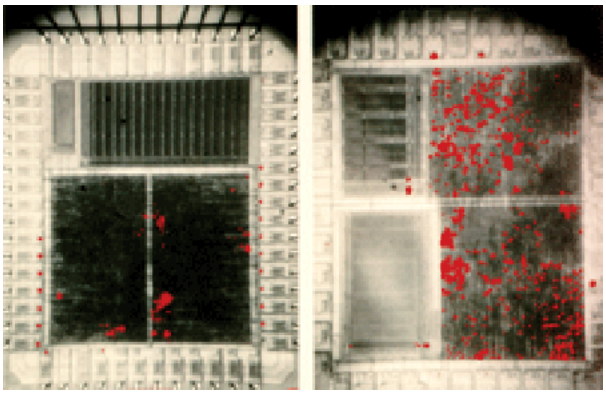
- As clock speeds increase, it becomes more difficult to get the clock signal to propagate to all parts of the chip fast enough for the circuit to operate properly.
- There is some upper limit to clock speed that depends on the material properties of the device.

Another Option:

- Eliminate the global clock.
- Use handshaking between the stages to handle data transfer.
- A circuit that does not utilize a synchronized clock is called an **ASYNCHRONOUS** circuit.

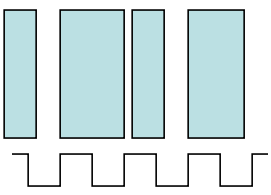
Synchronous Logic Comparison

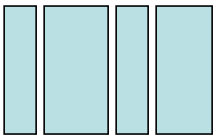
- Only active circuits run
- Entire chip runs



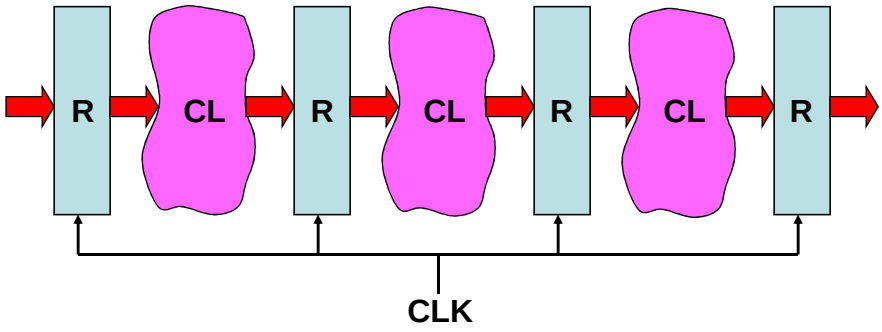
Synchronous Logic Comparison

- Worst case timing
  - Clock forces short functions to take same amount of time as long functions
- Best case timing: each module runs as fast as it can



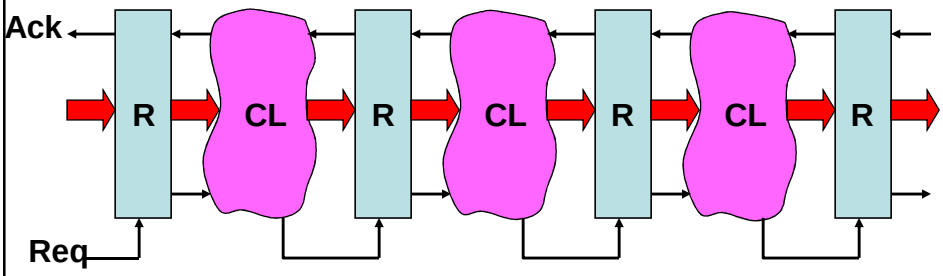


Synchronous circuit



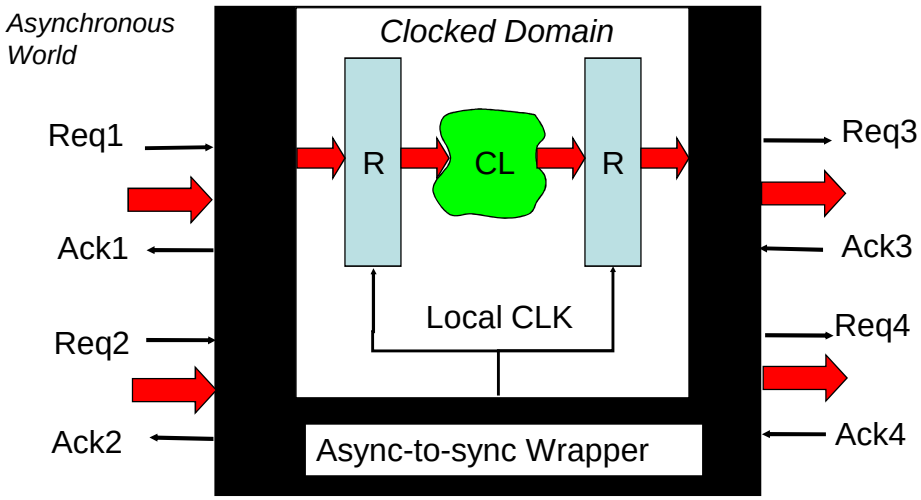
Implicit (global) synchronization between blocks
Clock period > Max Delay (CL + R)

Asynchronous circuit



Explicit (local) synchronization:
Req / Ack handshakes

Globally Async Locally Sync (GALS)



Clock disadvantages:

- Clock trees take up large amounts of space on die
- They consume large amounts of power
- Introduce noise
- Introduce delays

- Clock skew displaces useful work

  Tsynch = Tlogic + Tskew + Tflip-flop

  Tasynch = Tlogic + Thand-shake

Clock disadvantages ctnd.:

- Can take up to 40% power of consumption, 10% of chip area, and up to 50% of the performance
- Feature sizes decreasing, local clock speeds increasing
- Interconnects not keeping pace with transistor speed:
  - Past: Transistors limit speed
  - Future: Wires limit speed
- Clock skew tends to be proportional to interconnect delay

Benefits of Asynchronous Circuits

- High Speed -- Each stage can process as fast as possible, independent of a global clock.
- Low Power -- When system is idle, there is no clock driving the circuit.
- Modularity -- One design can easily be integrated into another design, without regard to how fast each section works.
- Design Time -- Less of the design time is spent making sure the timing assumptions of the circuit work at the given clock speed.

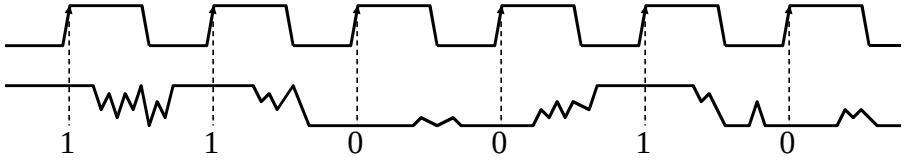
Disadvantages of Asynchronous Circuits

- Handshaking control can increase the overall circuit area.
- More difficult to design:
  - Most circuit designers learn only synchronous design.
  - Most CAD tools only support synchronous design.
  - Asynchronous circuit CAD tools still in early development.

Uses of Asynchronous Circuits

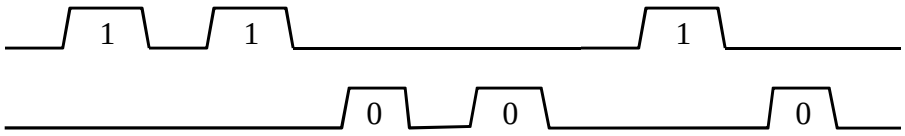
- Custom-designed circuits for high speed calculations.
- Modular designs that can be quickly changed for different applications.
- Low power uses, such as mobile phones and pagers.

Synchronous communication



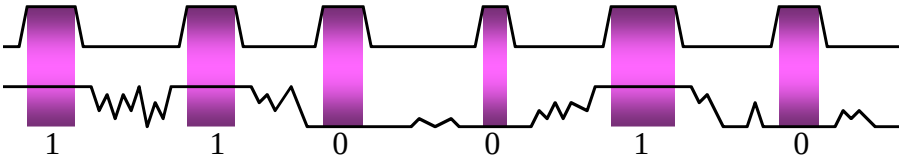
- Clock edges determine the time instants where data must be sampled
- Data wires may glitch between clock edges (set–up/hold times must be satisfied)
- Data are transmitted at a fixed rate (clock frequency)

Dual rail



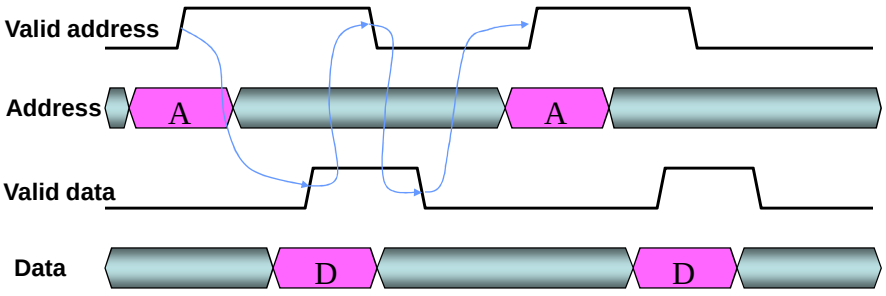
- Two wires with L(low) and H (high) per bit
  – “LL” = “spacer”, “LH” = “0”, “HL” = “1”
- *n*–bit data communication requires 2*n* wires
- Each bit is *self-timed*
- Other *delay-insensitive* codes exist (e.g. k-of-n) and event–based signalling (choice criteria: pin and power efficiency)

Bundled data



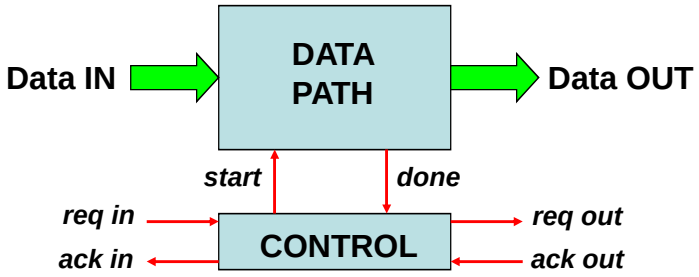
- Validity signal
  - Similar to an aperiodic local clock
- *n*–bit data communication requires *n*+1 wires
- Data wires may glitch when no valid
- Signaling protocols
  - level sensitive (latch)
  - transition sensitive (register): 2–phase / 4–phase

Example: memory read cycle



- Transition signaling, 4-phase

Asynchronous modules

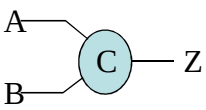


- **Signaling protocol:**

  reqin+ start+ [*computation*] done+ reqout+ ackout+ ackin+
  reqin- start- [*reset*] done- reqout- ackout- ackin-

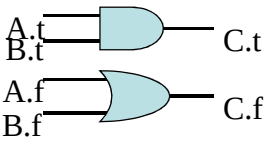
  (more concurrency is also possible)

Elements for Asynchronous Design

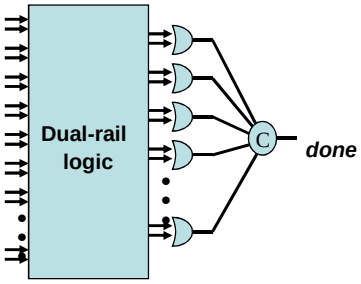


A	B	Z^+
0	0	0
0	1	Z
1	0	Z
1	1	1

C-element

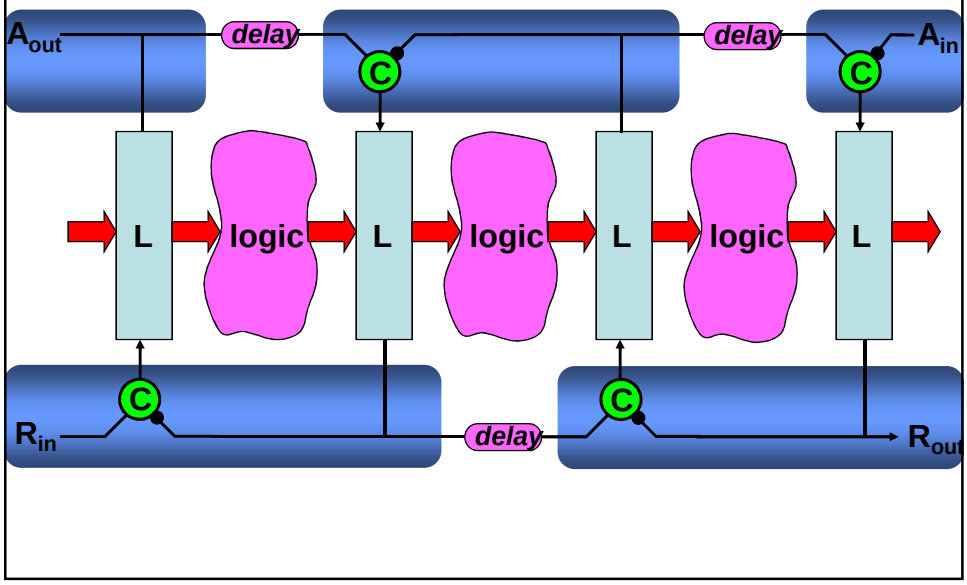


Dual-rail AND gate

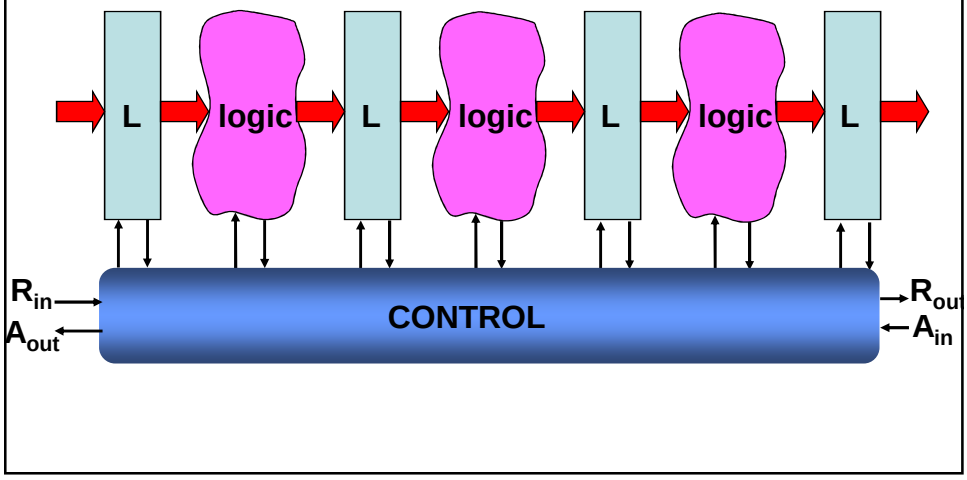


Completion detection circuit

Micropipelines (Sutherland 89)

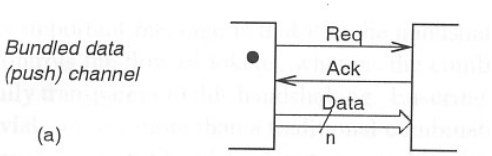


Data-path / Control

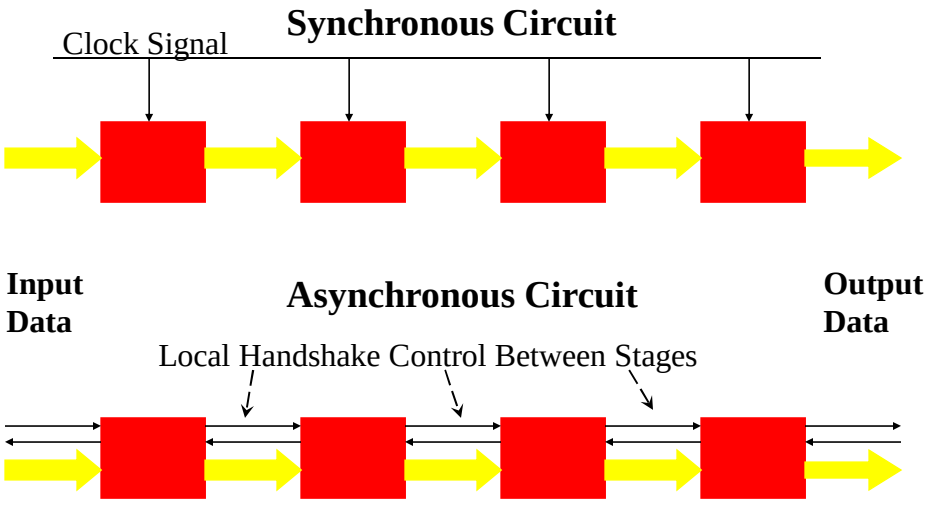


Handshake Communication

- Most common control protocol is the Request/Acknowledge protocol
- The sender issues a Request to the receiver before sending the data. The receiver replies by sending an Acknowledge to the sender. Then the sender sends the data.
- If the sender initiates the data transfer, the transfer channel is a *push-channel*. On the other hand if the receiver initiates the transfer, the channel is a *pull-channel*.



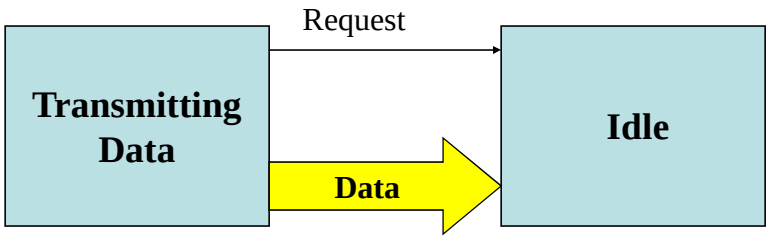
Synchronous v. Asynchronous



How Handshaking Works

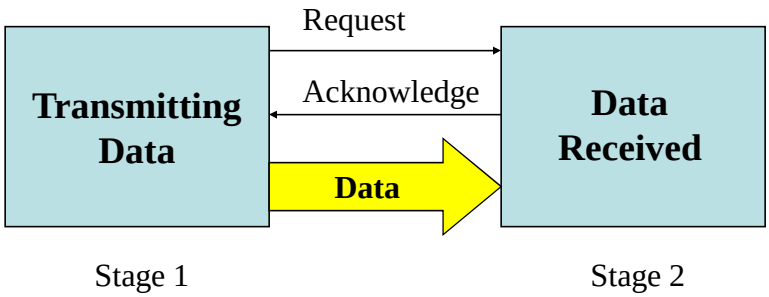
Stage 1 Stage 2

How Handshaking Works



Stage 1 Stage 2

How Handshaking Works



How Handshaking Works

Handshake Protocols

- Two Phase
  - The sender sends data and produces a Req event
  - The receiver absorbs data and produces an Ack event
  - Edge sensitive

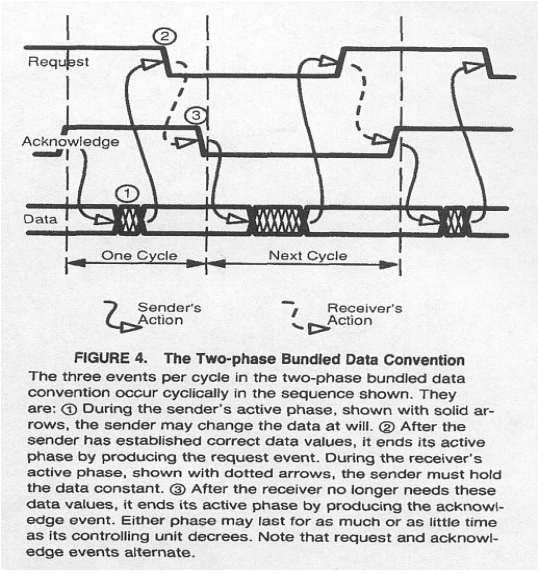
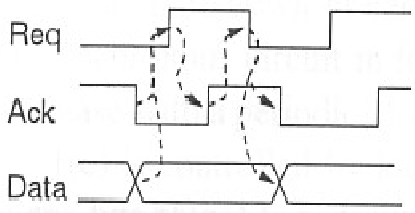


FIGURE 4. The Two-phase Bundled Data Convention
The three events per cycle in the two-phase bundled data convention occur cyclically in the sequence shown. They are: ① During the sender's active phase, shown with solid arrows, the sender may change the data at will. ② After the sender has established correct data values, it ends its active phase by producing the request event. During the receiver's active phase, shown with dotted arrows, the sender must hold the data constant. ③ After the receiver no longer needs these data values, it ends its active phase by producing the acknowledge event. Either phase may last for as much or as little time as its controlling unit decrees. Note that request and acknowledge events alternate.

Handshake Protocols

Four Phase
- The sender issues data and sets Req to high
- The receiver absorbs the data and sets Ack to high
- The sender responds by setting Req to low
- The receiver acknowledges by setting Ack to low
- Level sensitive



Conclusion

- Asynchronous design provides delay-independent circuits.
- Synthesis from STGs can be fully automated.
- Great reduction in power
- No affect in functionality