L12: I/O Systems

Name: ________

ID: ________

Synchronous and Asynchronous Buses

Synchronous Bus (e.g., processor-memory buses)

- Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
-
-
-

Asynchronous Bus (e.g., I/O buses)

- It is not clocked, so requires a handshaking protocol and additional control lines (ReadReq, Ack, DataRdy)
-
-
-
-

Synchronous and Asynchronous Buses

Synchronous Bus (e.g., processor-memory buses)

- Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
- ☺ Involves very little logic and can run very fast
-
-

Asynchronous Bus (e.g., I/O buses)

- It is not clocked, so requires a handshaking protocol and additional control lines (ReadReq, Ack, DataRdy)
-
-
-
-

Synchronous and Asynchronous Buses

Synchronous Bus (e.g., processor-memory buses)

- Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
- ☺ Involves very little logic and can run very fast
- ☹ Every device communicating on the bus must use same clock rate
- ☹ To avoid clock skew, they cannot be long if they are fast

Asynchronous Bus (e.g., I/O buses)

- It is not clocked, so requires a handshaking protocol and additional control lines (ReadReq, Ack, DataRdy)
-
-
-
-

Synchronous and Asynchronous Buses

Synchronous Bus (e.g., processor-memory buses)

- Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
- ☺ Involves very little logic and can run very fast
- ☹ Every device communicating on the bus must use same clock rate
- ☹ To avoid clock skew, they cannot be long if they are fast

Asynchronous Bus (e.g., I/O buses)

- It is not clocked, so requires a handshaking protocol and additional control lines (ReadReq, Ack, DataRdy)
- ☺ Can accommodate a wide range of devices and device speeds
- ☺ Can be lengthened without worrying about clock skew
-
-

Synchronous and Asynchronous Buses

Synchronous Bus (e.g., processor-memory buses)

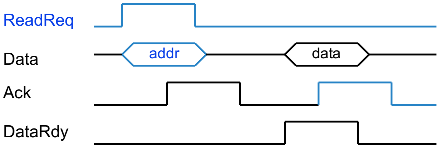
- Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
- ☺ Involves very little logic and can run very fast
- ☹ Every device communicating on the bus must use same clock rate
- ☹ To avoid clock skew, they cannot be long if they are fast

Asynchronous Bus (e.g., I/O buses)

- It is not clocked, so requires a handshaking protocol and additional control lines (ReadReq, Ack, DataRdy)
- ☺ Can accommodate a wide range of devices and device speeds
- ☺ Can be lengthened without worrying about clock skew
- ☹ Slow(er)
- ☹ More complex due to handshaking protocol

Asynchronous Bus Handshaking Protocol

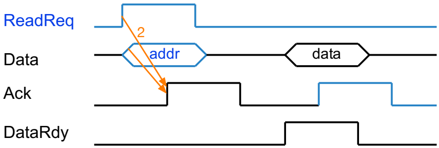
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2.
3.
4.
5.
6.
7.
8.

Asynchronous Bus Handshaking Protocol

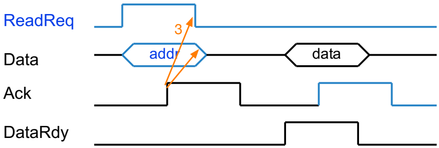
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3.
4.
5.
6.
7.
8.

Asynchronous Bus Handshaking Protocol

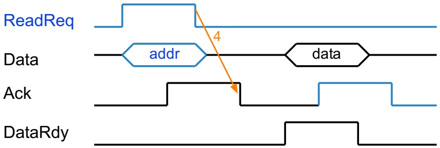
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3. I/O device sees `Ack` and releases the `ReadReq` and data lines
4.
5.
6.
7.
8.

Asynchronous Bus Handshaking Protocol

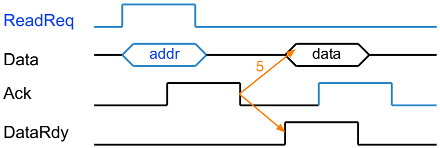
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3. I/O device sees `Ack` and releases the `ReadReq` and data lines
4. Memory sees `ReadReq` go low and drops `Ack`
5.
6.
7.
8.

Asynchronous Bus Handshaking Protocol

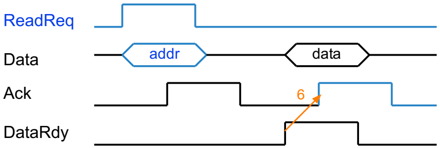
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3. I/O device sees `Ack` and releases the `ReadReq` and data lines
4. Memory sees `ReadReq` go low and drops `Ack`
5. When memory ready, putting data on data lines & raises `DataRdy`
6.
7.
8.

Asynchronous Bus Handshaking Protocol

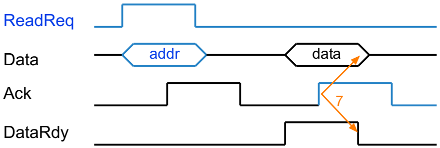
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3. I/O device sees `Ack` and releases the `ReadReq` and data lines
4. Memory sees `ReadReq` go low and drops `Ack`
5. When memory ready, putting data on data lines & raises `DataRdy`
6. I/O device sees `DataRdy`, reads data from data lines & raises `Ack`
7.
8.

Asynchronous Bus Handshaking Protocol

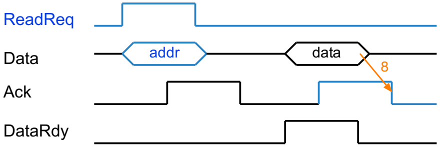
Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3. I/O device sees `Ack` and releases the `ReadReq` and data lines
4. Memory sees `ReadReq` go low and drops `Ack`
5. When memory ready, putting data on data lines & raises `DataRdy`
6. I/O device sees `DataRdy`, reads data from data lines & raises `Ack`
7. Memory sees `Ack`, releases data lines, and drops `DataRdy`
8.

Asynchronous Bus Handshaking Protocol

Example: data from Memory to I/O devices



1. I/O device requests by raising `ReadReq` & putting addr on the data lines
2. Memory sees `ReadReq`, reads addr from data lines, and raises `Ack`
3. I/O device sees `Ack` and releases the `ReadReq` and data lines
4. Memory sees `ReadReq` go low and drops `Ack`
5. When memory ready, putting data on data lines & raises `DataRdy`
6. I/O device sees `DataRdy`, reads data from data lines & raises `Ack`
7. Memory sees `Ack`, releases data lines, and drops `DataRdy`
8. I/O device sees `DataRdy` go low and drops `Ack`