

Bus protocols

Digital Systems M, Module 1
Stefano Mattoccia, Università di Bologna



Communication protocols

Standard communication protocols

- ✓ Protocols for peripherals and memory devices
- ✓ 3 out of 4 described next proposed by ARM
- ✓ Industrial standards
- ✓ each one tailored for a specific application domain
- ✓ all available on the Zynq platform

- I2C
- ARM AXI
- ARM AXI Lite
- ARM AXI Stream (address less)

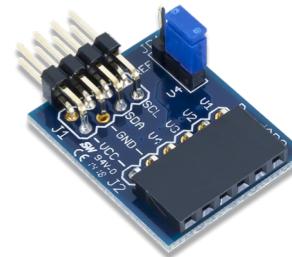
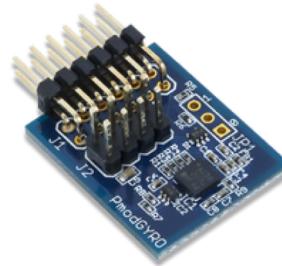
I²C (I²C) protocol

I²C (Inter-integrated Circuit) was initially proposed by Philips in the 80's.

Widespread diffusion since then in many contexts (cameras, sensors, CPU, etc) due to its simplicity:

- serial protocol
- only 2 wires: SCL (clock) e SDA (data)
- initial frequency 100 KHz
- current standard frequency 400 KHz (often 1+ MHz)
- adopted by most devices not only Zynq
- simple and lightweight
- sometimes named differently (eg, for imaging sensor OV 7670 referred to as SCCB)
- SPI is a similar yet slightly more complex serial protocol

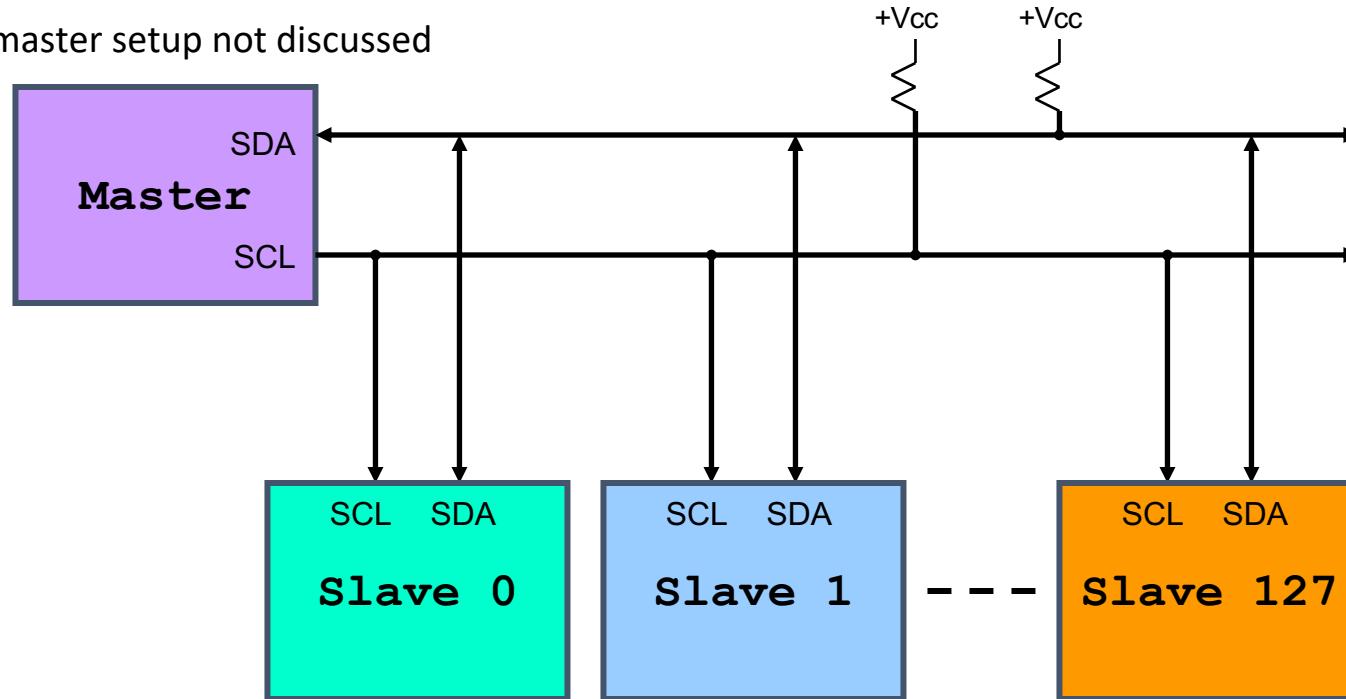
Bus protocols



.....

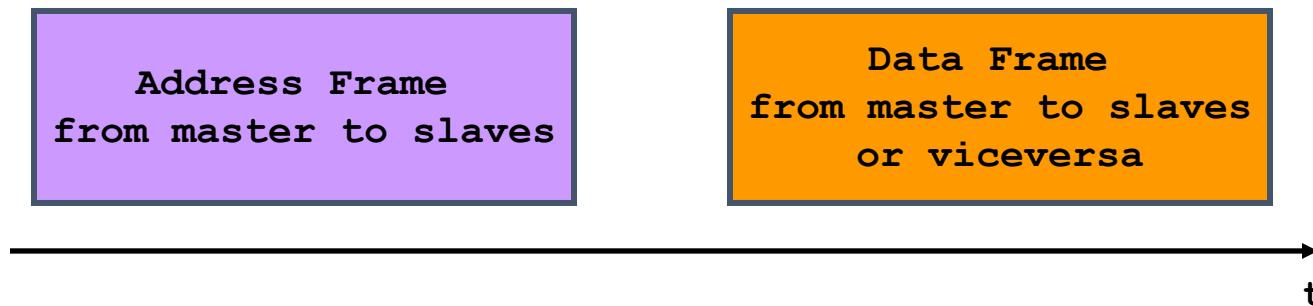
Bus protocols

- The master always starts the communication
- At most 128 slave devices (actually, much less in most practical applications)
- In most cases the address is assigned by the manufacturer and can't be changed
- Multi-master setup not discussed

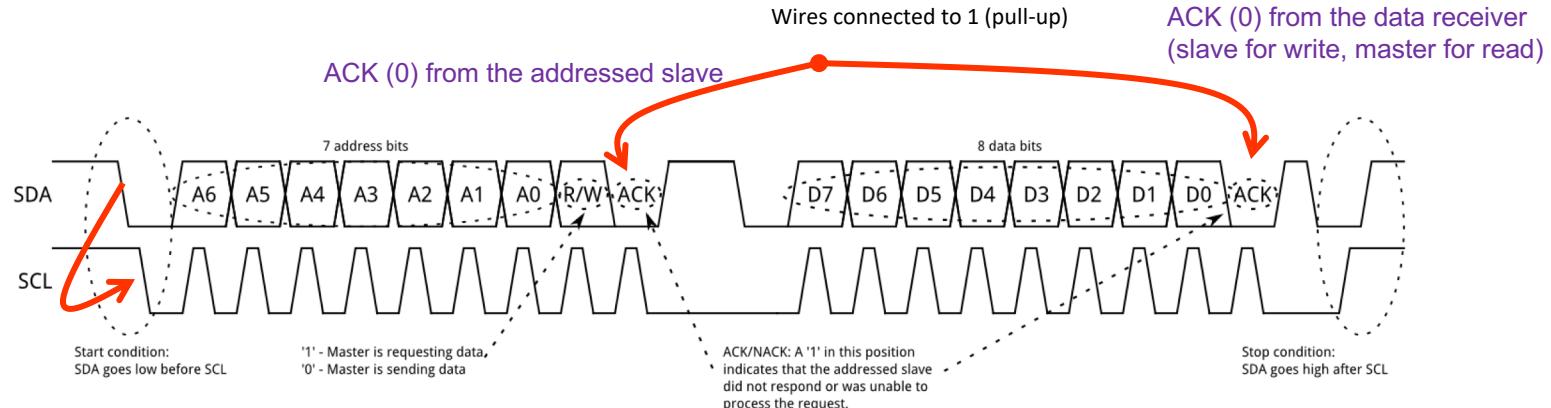


Bus protocols

- Each bus cycle consists of a sequence of two frames:
 - 1) the master emits the address of the slave involved in the communication
 - 2) the master specifies if it is a read or write cycle
- Data Frame
 - a) for write: the master sends data to the slave device
 - b) for read: the master reads from the slave device



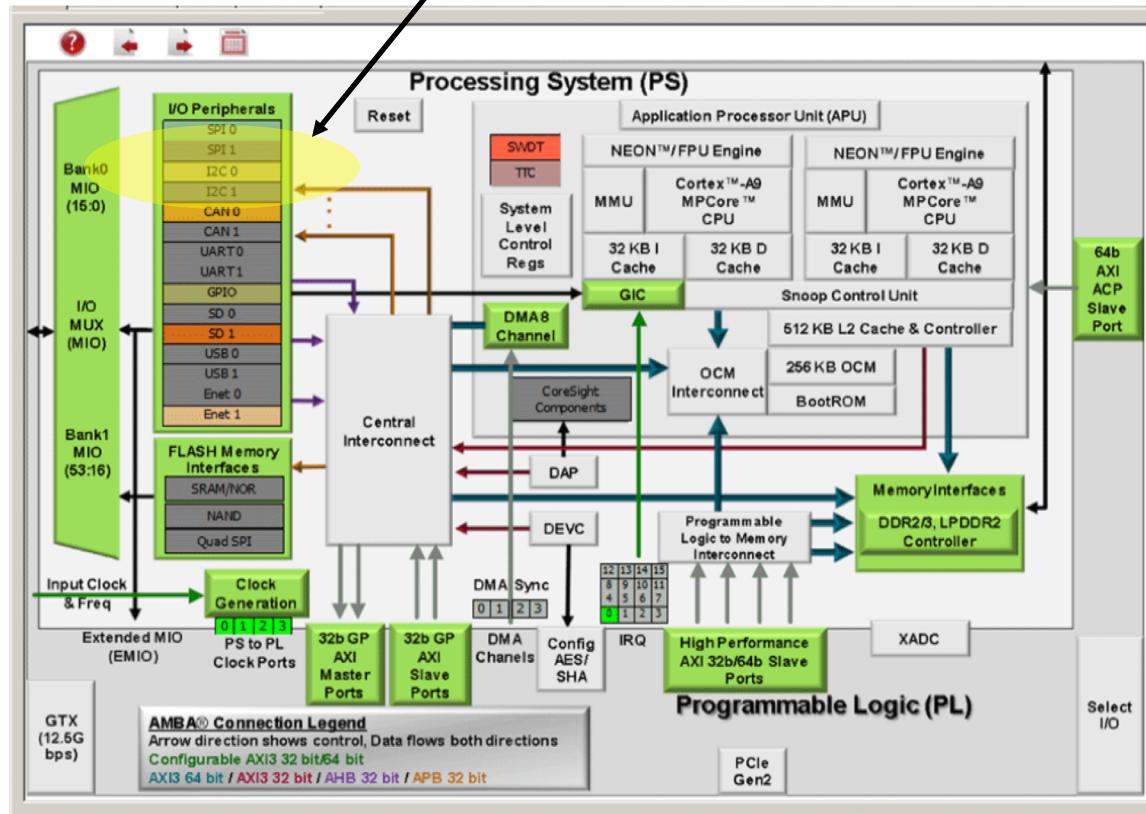
Bus protocols



**Address Frame
from master to slaves**

**Data Frame
from master to slaves
or viceversa**

Bus protocols

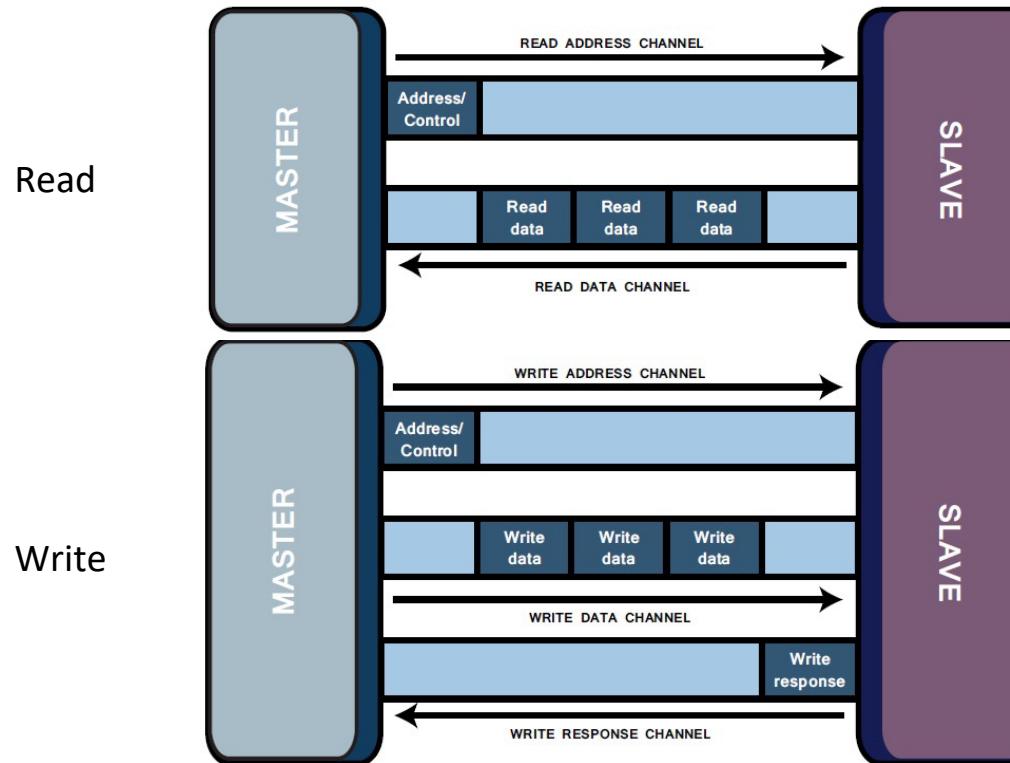


Zynq: on-chip communication protocols

- **AXI4**: suited for high performance **memory mapped** communications. It handles up to 256 transfers providing a single address (*burst transfer*).
- **AXI4-Lite**: enables a single (no burst) transfer for **memory mapped** devices.
- **AXI4-Stream**: suited for high performance data stream transfer with devices not requiring conventional addressing (i.e. **not memory mapped**). Extremely fast and lightweight, often deployed to manage image/signal streams

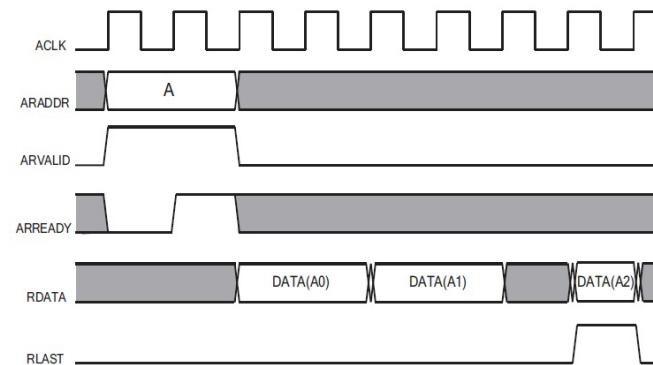
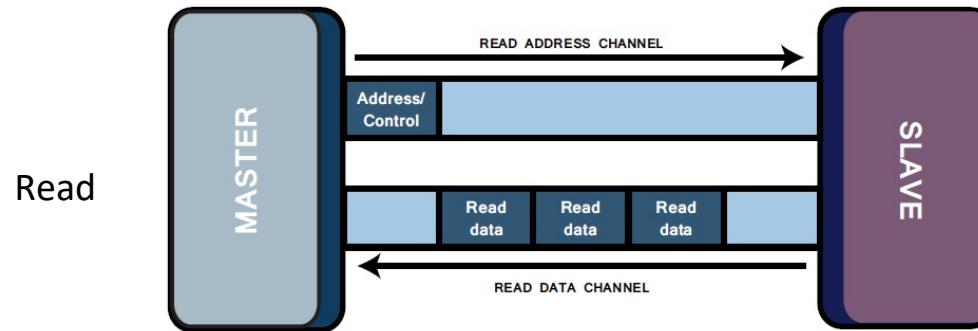
Bus protocols

AXI: communication with memory mapped devices



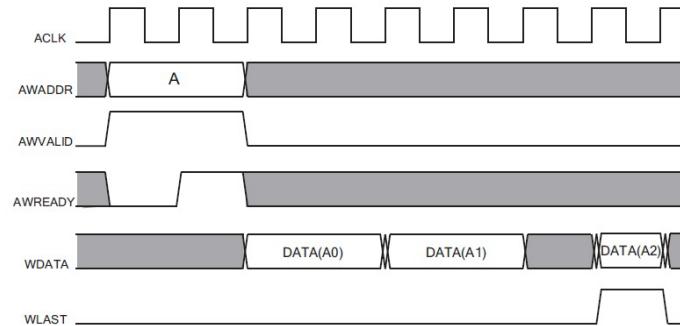
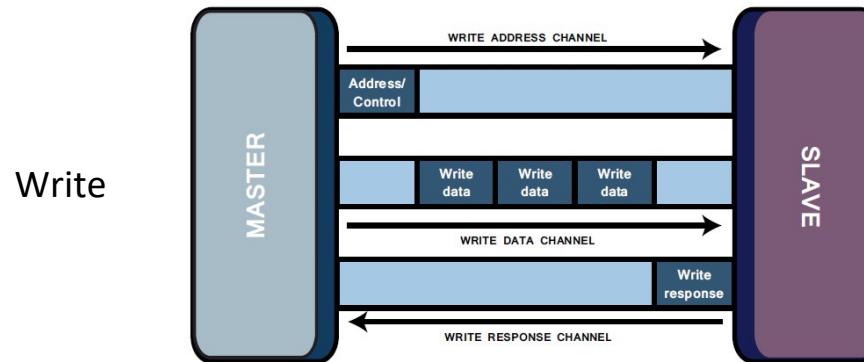
Bus protocols

AXI: burst read



Bus protocols

AXI: burst write



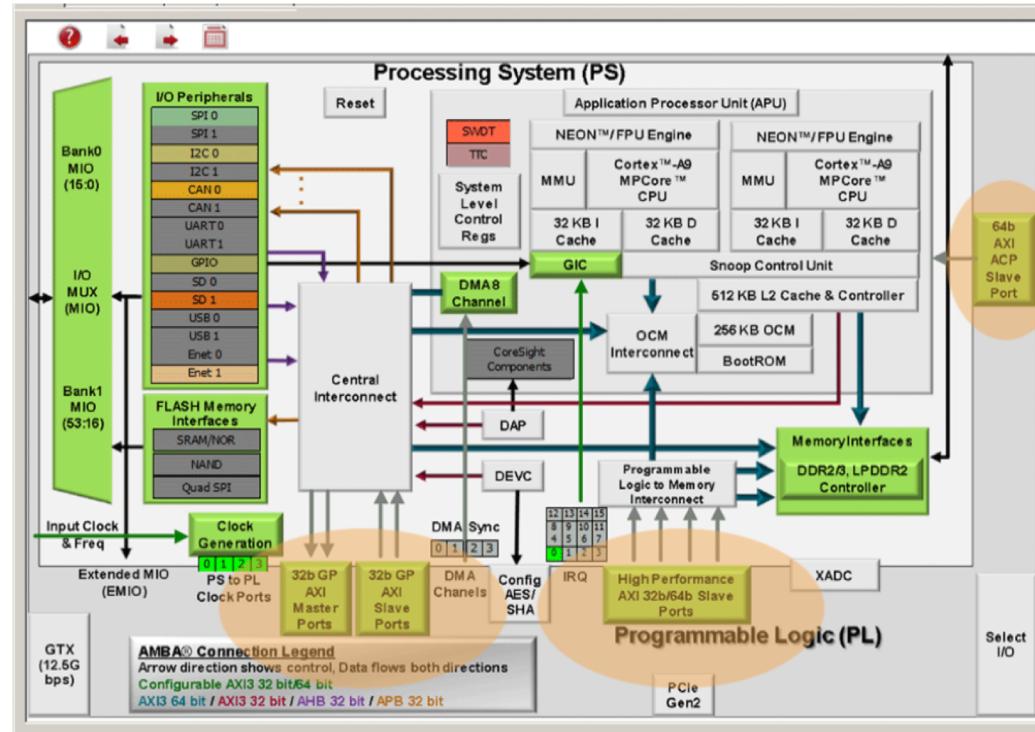
Bus protocols

Zynq interface

Interface Name	Interface Description	Master	Slave
M_AXI_GP0	General Purpose (AXI_GP)	PS	PL
M_AXI_GP1		PS	PL
S_AXI_GP0	General Purpose (AXI_GP)	PL	PS
S_AXI_GP1		PL	PS
S_AXI_ACP	Accelerator Coherency Port (ACP), cache coherent transaction	PL	PS
S_AXI_HP0	High Performance Ports (AXI_HP) with read/write FIFOs. (Note that AXI_HP interfaces are sometimes referred to as AXI Fifo Interfaces, or AFIs).	PL	PS
S_AXI_HP1		PL	PS
S_AXI_HP2		PL	PS
S_AXI_HP3		PL	PS

Bus protocols

Zynq architecture and AXI4 ports



Bus protocols

AXI4 and AXI4-Lite 1/4

- AXI4 e AXI4-Lite enable Zynq to exchange data between the *Processing System* and *Programmable Logic* via *General Purpose* or *High Performance* ports
- AXI has 32 bit addresses and configurable parallelism (32, 64, 128, 256, 512 e 1024 bit), with a significant impact on performance and resources
- Clock frequency can be set by the designer
- Further details available here:

<http://www.arm.com/products/system-ip/amba/amba-open-specifications.php>

<http://www.xilinx.com/ipcenter/axi4.htm>

Bus protocols

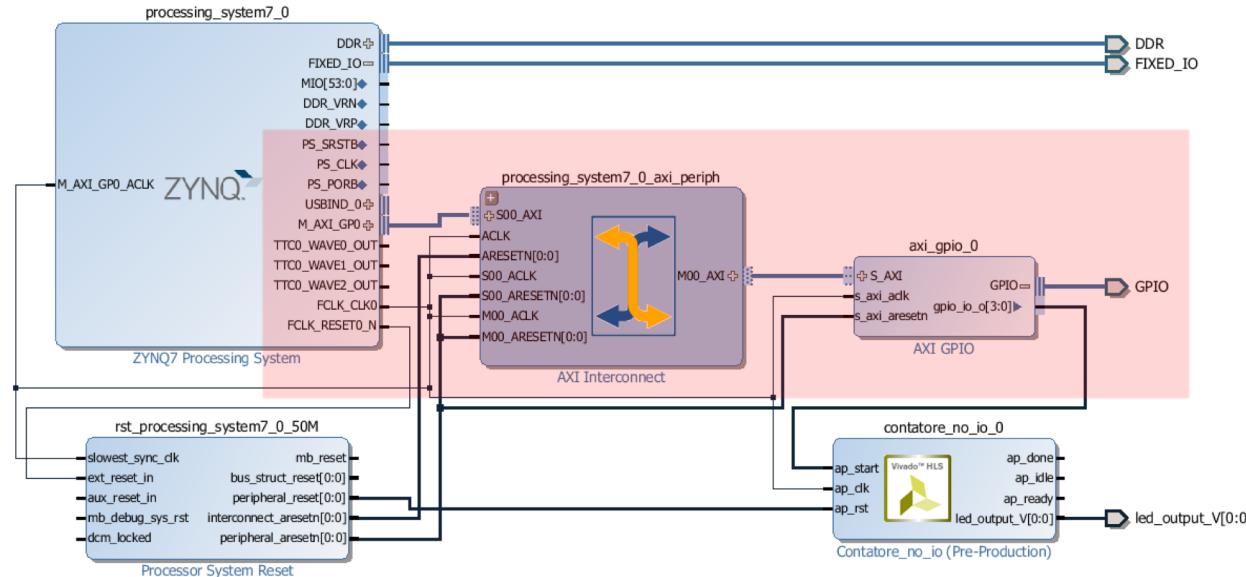
AXI4 and AXI4-Lite 2/3

- Read Address Channel
- Write Address Channel
- Read Data Channel
- Write Data Channel
- Write Response Channel
- Bi-directional communication channels (for address, data and control)
- Configurable data parallelism
- Zynq: AXI4 ports HP_0,HP_1,HP_2,HP_3 are slave
while AXI-lite ports can be configured either as master or slave

Bus protocols

AXI4 and AXI4-Lite 3/3

- Data transfers occur between a master ad a slave
- A general purpose component (referred to as *interconnect*) easily allows to manage multiple connections



Bus protocols

AXI Stream

- The communication between a master and a slave occurs without addressing (not *memory mapped*)
- Master e slave sincronize thirself through *handshake*
- Unbounded number of transfers
- Data parallelism can be configured
- Of course, random access to data is not allowed (it would require and address)

