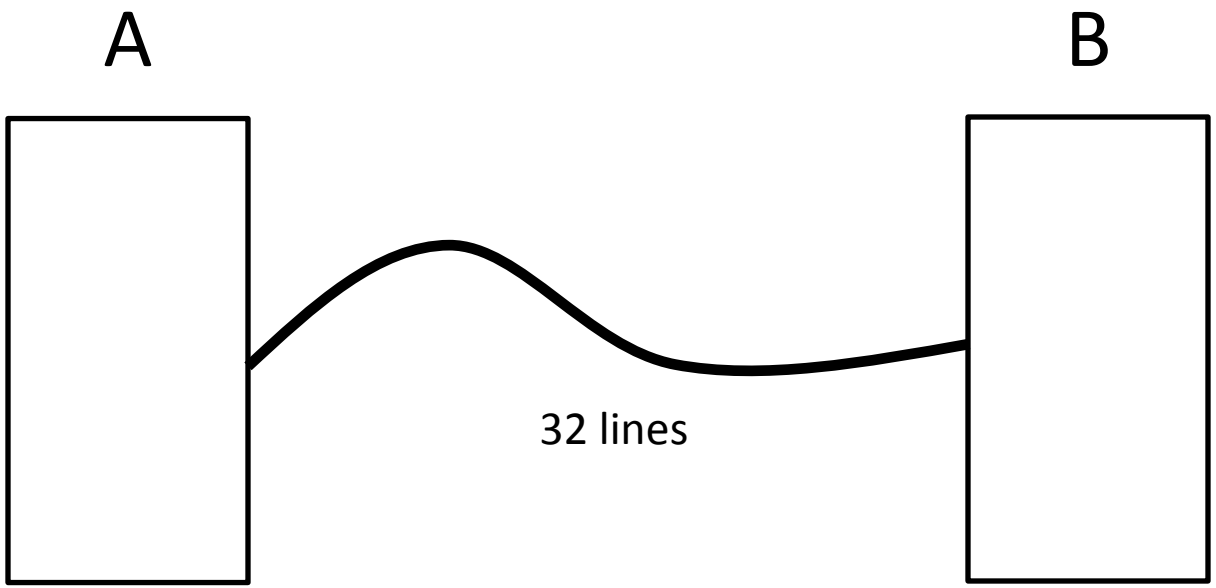
VERY basic VME

E.J. – 10/18

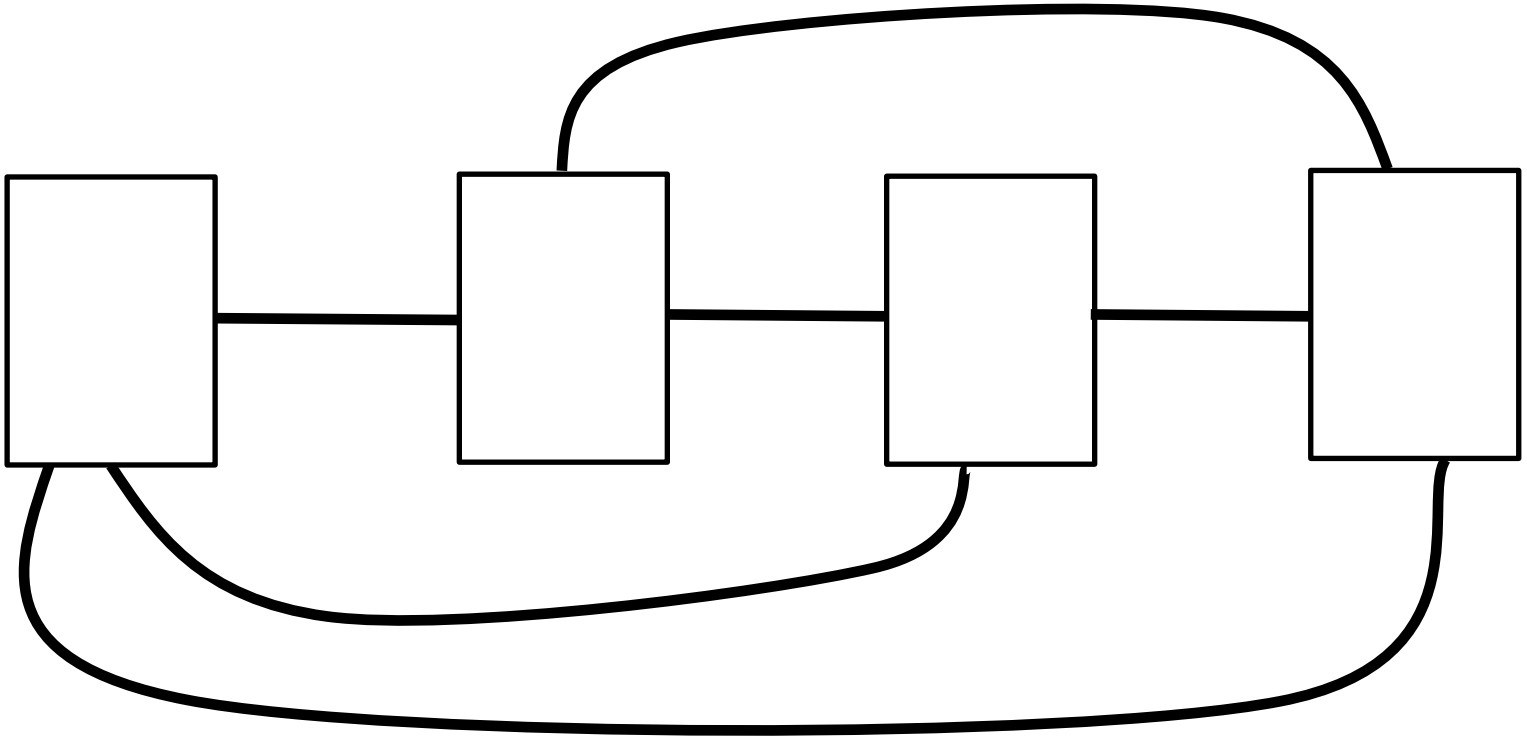
Parallel Data Transfer

- Until recently (~2000) the cheapest and easiest way to transfer data between 2 points at high rate was using a parallel data line architecture



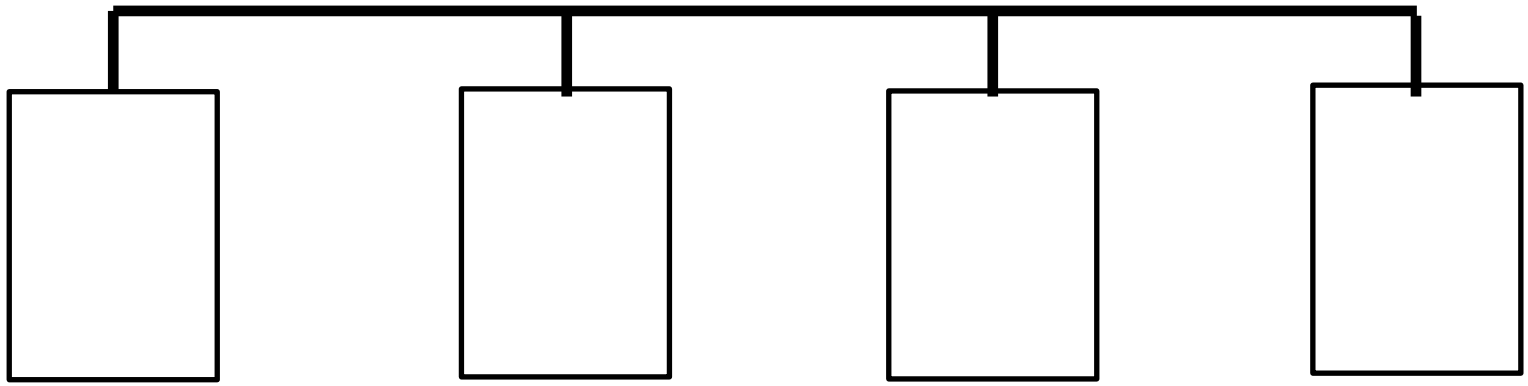
All 32 bits of data are transmitted simultaneously

Problems with more than 2 points



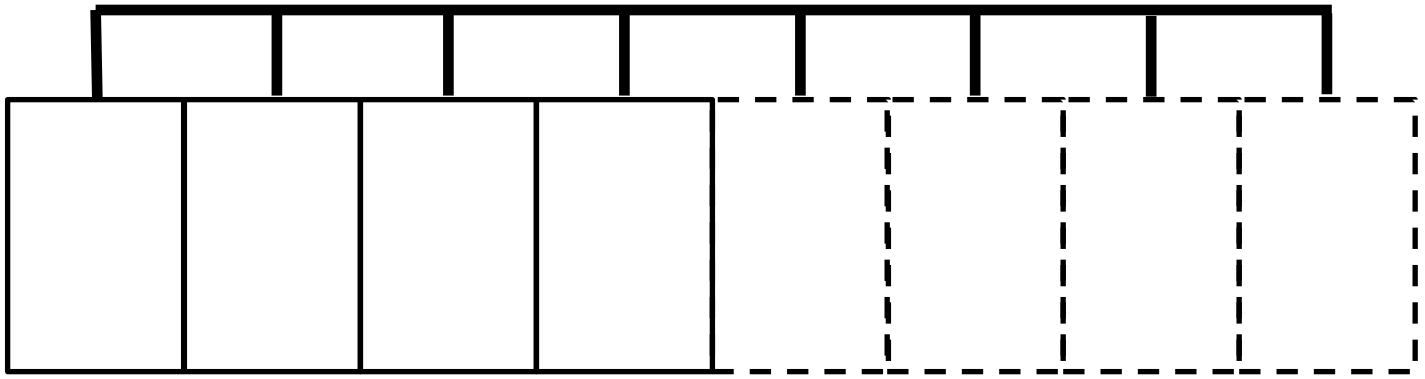
- Potentially lots of cables and connectors
- System doesn’t scale well – must know how many points

Bus Architecture



- All data units share COMMON SIGNAL LINES
- Only a SINGLE PAIR of data units can transfer data at a given time (Multiplexing in time)

Bus/Crate Architecture

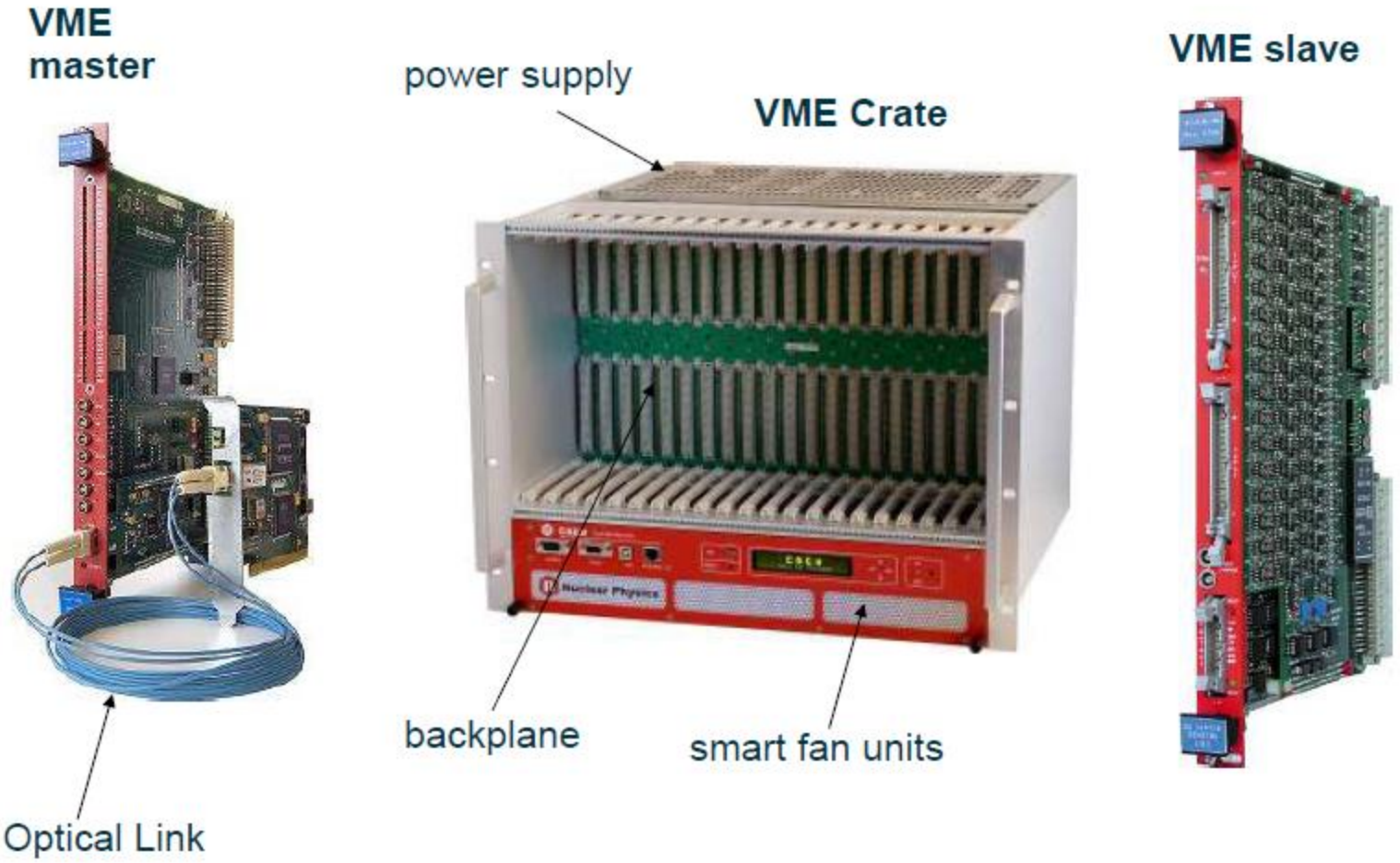


- Data units (modules) occupy slots in a chassis
- Common data lines are on a printed circuit board (BACKPLANE)
- Multiple slots (~20) allow for dense packaging and system expansion
- Modules share common power supply and cooling unit (saves $$)
- Allows for STANDARDIZATION (e.g. VME)
- With standardization we can mix modules from different vendors and custom designed modules

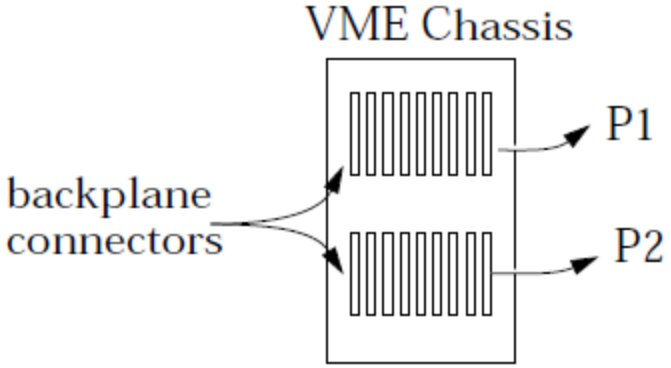
Introduction to VME

- VME stands for VERSA-Module Euro card introduced in 1981 for industrial, commercial and military applications.
- Electrical and mechanical specifications are defined by the standard.
- VME bus is a master-slave computer architecture.
- The signaling scheme is asynchronous, meaning that the transfer is not tied to the timing of a bus clock.
- **VITA** (VME International Trade Association) is the organisation whose purpose is to promote and develop the VME

VME Components



VME Backplane



The VMEbus is designed for flexibility to accommodate varying needs:

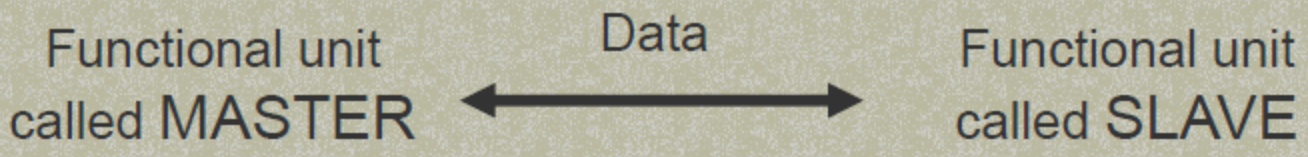
P1/J1	24 bit addressing, 16 bit data, plus control lines
P2/J2	Expands capability to 32 bit addressing and 32 bit data

Applications

- Industrial Control
- Military
- Aerospace
- Transportation
- Telecom
- Simulation
- Medical
- High energy physics
- General business

VME bus features

- Master / slave architecture



- Since multiple Masters can reside on the bus, it is a MULTIPROCESSING bus
- Can take from 1 to 21 Masters

VME

- Data transfers are between a ***pair*** of modules.
- A ***master*** module initiates the data transfer.
- The ***master*** selects the target or ***slave*** module with the address it drives on the bus. Each ***slave*** module has a unique address*.
- A ***slave*** will participate in the data transfer when it recognizes its address. All other ***slave*** modules will ignore the ***master***.
- The direction of the transfer (write, read) is from the ***master's*** point of view:
  - **WRITE**: the ***master*** transfers data to the ***slave***
  - **READ**: the ***master*** retrieves data from the ***slave***
- All of the data acquisition modules we design at JLab are ***slaves***.

(* actually a unique range of addresses)

Basic Addressing

- A module (slave) has a unique *range* of addresses it will respond to.
- For VME the basic unit of storage is the byte (8 bits). Every byte has a unique address.
- A VME slave module typically has many bytes of storage. For convenience the byte storage within the module is arranged contiguously.
- Example - For 24-bit addressing with 6 modules:
  - Module 1 – address range 0x100000 – 1000FF (256 bytes storage)
  - Module 2 – address range 0x100100 – 1001FF (256 bytes storage)
  - Module 3 – address range 0x200000 – 200FFF (4096 bytes storage)
  - Module 4 – address range 0x201000 – 201FFF (4096 bytes storage)
  - Module 5 – address range 0x202000 – 202FFF (4096 bytes storage)
  - Module 6 – address range 0x300000 – 30000F (16 bytes storage)

- NONE of the addresses from the modules can overlap.
- Modules 1 – 6 could be placed in any order and in any of the 21 VME slots. It is the address not the physical location (slot) that identifies the module.

VME Address Map for Example

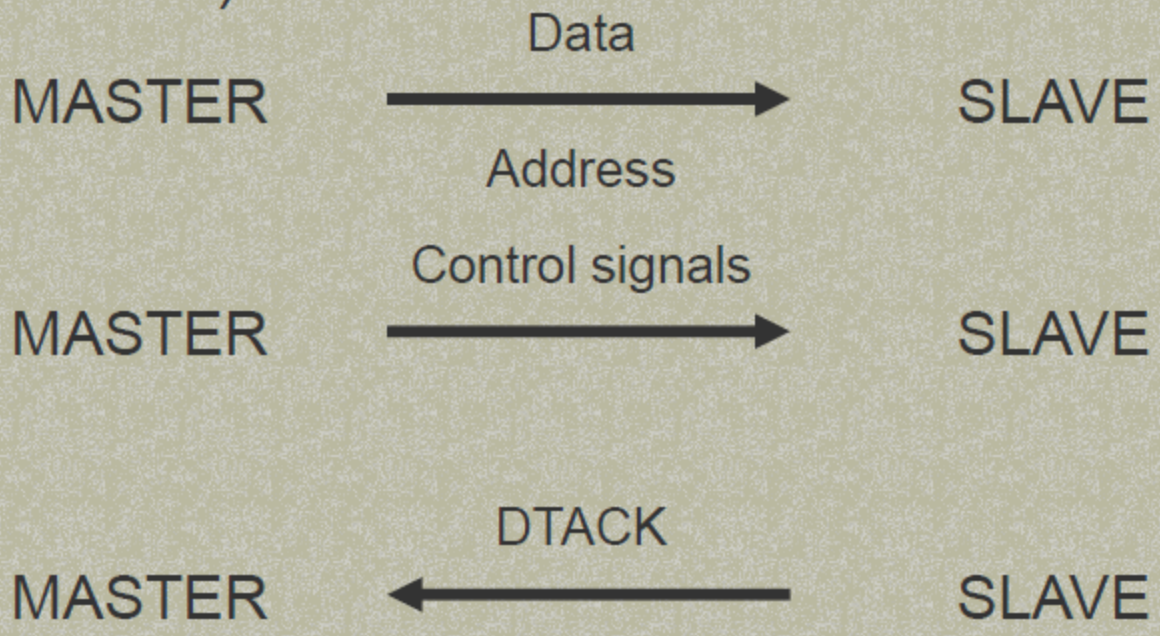
- 000000 unused
- 0FFFFF
- **100000 Module 1**
- **1000FF**
- **100100 Module 2**
- **1001FF**
- 100200 unused
- 1FFFFF
- **200000 Module 3**
- **200FFF**
- **201000 Module 4**
- **201FFF**
- **202000 Module 5**
- **202FFF**
- 203000 unused
- 2FFFFF
- **300000 Module 6**
- **30000F**
- 300010 unused
- FFFFFF

4 Byte Groups

- It is often convenient to organize storage in a module as 32-bit words (4 bytes).
- VME allows 32-bit data transfers to 4 byte groups.
- For Module 1 in our example the 4 byte groups are:
  - **100000**, 100001, 100002, 100003
  - **100004**, 100005, 100006, 100007
  - **100008**, 100009, 10000A, 10000B
  - **10000C**, 10000D, 10000E, 10000F
  - **100010**, 100011, 100012, 100013
  - **100014**, 100015, 100016, 100017
  - **100018**, 100019, 10001A, 10001B
  - **10001C**, 10001D, 10001E, 10001F,
  - etc.
- A 4 byte group is accessed using the 1st address of the group (e.g. **100008**) along with a specific pattern of control signals (DS0*, DS1*, LWORD*, A1) that identify the transfer as 4 bytes long.

VME bus features

- Asynchronous bus – there is no central synchronization clock (uses handshaking protocol)



- Advantages?

Signal Ordering

- VME bus has no clock.
- Each operation is a sequence of steps.
- A transition between steps is the rising/falling edge of a single control signal.
  - eg. Most operations start on the ~~1 → 0 transition~~ of the address strobe line.

(HIGH to LOW signal transition)

Signals that are asserted LOW are indicated with a *
e.g. DS*, DTACK*

VME Operations

- VME bus supports many different operations. We will talk about the three most common.

  Read Master takes data from Slave

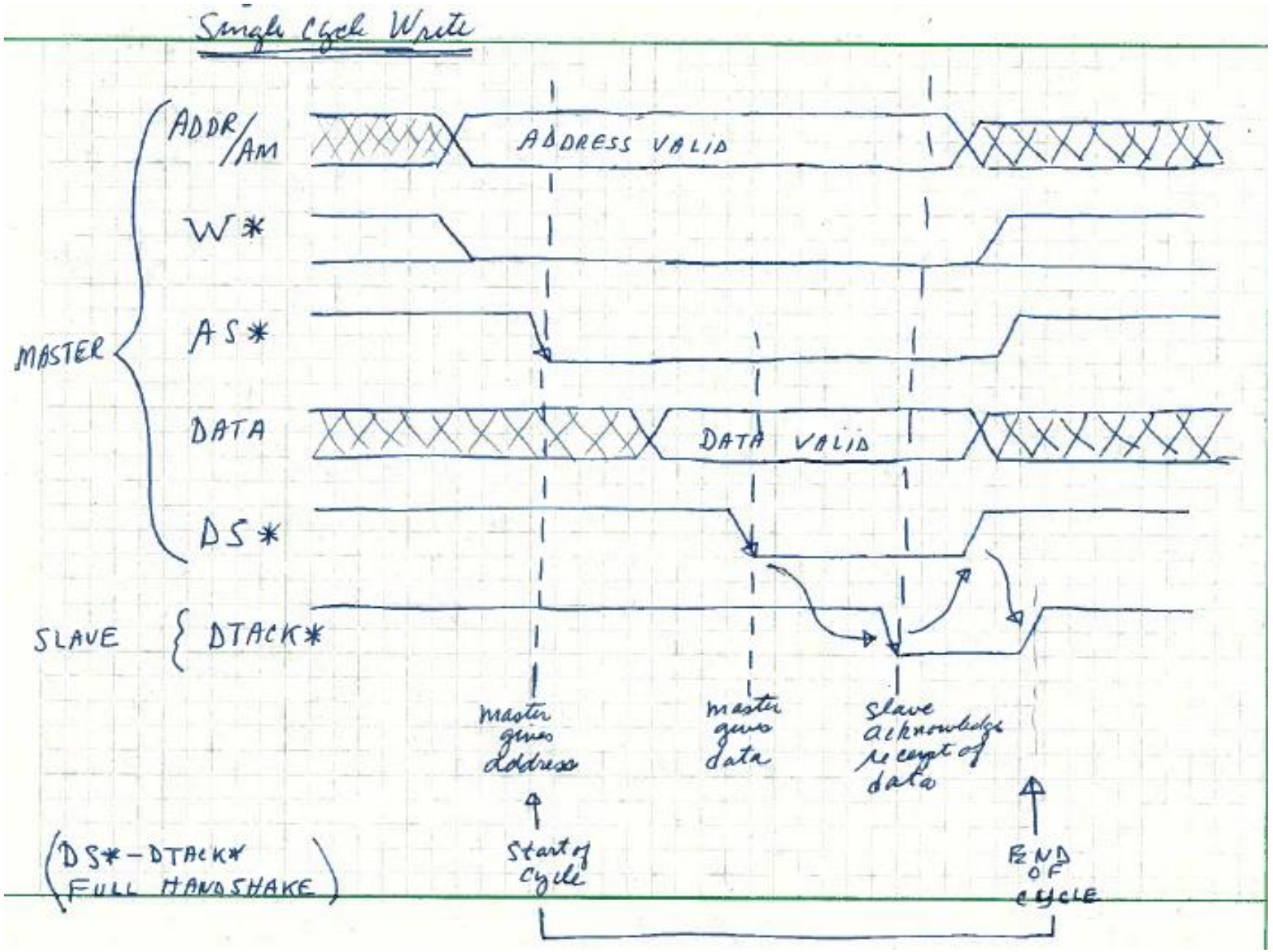
  Write Master pushes data to Slave

  ~~Interrupt Slave requests service from Master~~

- The following slides describe the single cycle (data mode) read/write operations. Block mode will not be discussed.

Single Cycle Write

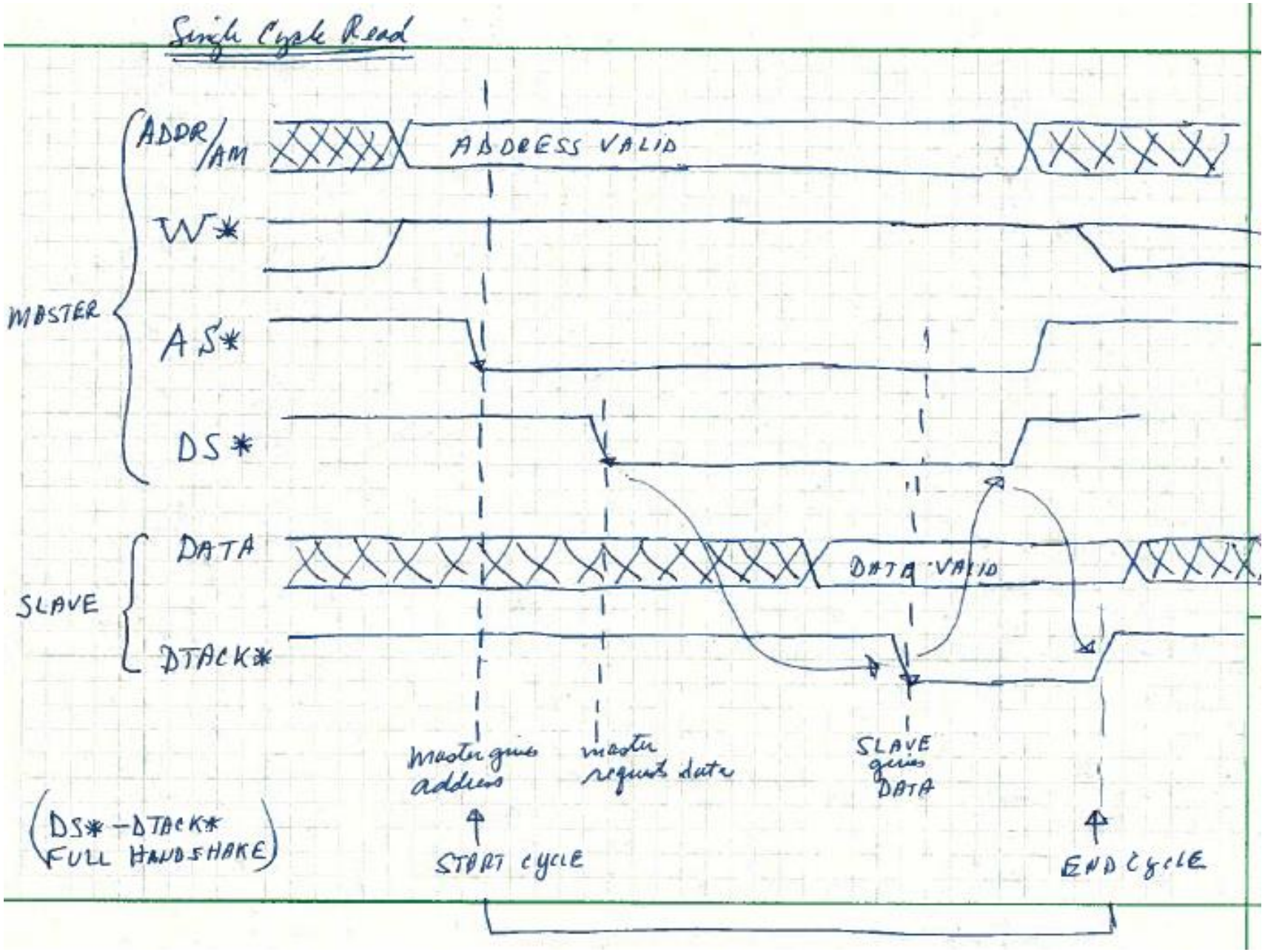
- Operation to write data to a single address.
- Address modifiers: A16nD, A24nD, A32nD (also sup. modes)
- Data widths: 8-bit, 16-bit, or 32-bit
- Master signals
  - Address Strobe AS*
  - Address and Address modifer ADDR(31-0), AM(5-0)
  - Data strobes DS0*, DS1*
  - Data DATA(31-0)
- Slave signals:
  - Data Acknowledge DTACK*


Single Cycle Write
ADDR/AM
ADDRESS VALID
W*
AS*
MASTER
DATA
DATA VALID
DS*
SLAVE
DTACK*
master gives address
master gives data
slave acknowledges receipt of data
Start of cycle
END OF CYCLE
(DS*-DTACK* FULL HANDSHAKE)

Single Cycle Read

- Operation to read data from a single address.
- Address modifiers: A16nD, A24nD, A32nD, A16sD, A24sD, A32sD
- Data widths: 8-bit, 16-bit, or 32-bit

- Master signals
  - Address Strobe AS*
  - Address and Address modifer ADDR(31-0), AM(5-0)
  - Data strobes DS0*, DS1*
- Slave signals:
  - Data Acknowledge DTACK*
  - Data DATA(31-0)


Single Cycle Read
ADDR/AM
ADDRESS VALID
W*
MASTER
AS*
DS*
DATA
DATA VALID
SLAVE
DTACK*
master gives address
master request data
SLAVE gives DATA
(DS* - DTACK* FULL HANDSHAKE)
START cycle
END cycle

VME bus features

- Addressing
  - Provides variety of address spaces and data widths – Dynamic address and data sizing
  - Makes no distinction between IO space and Memory space
  - Uses three address spaces
    - 16-bit (A16)
    - 24-bit (A24)
    - 32-bit (A32)
  - 6 bit address modifier code is used to distinguish between these address spaces

VME bus features

- Data Transfer
  - Provides variety of data widths – Dynamic data sizing
  - Data transfer sizes can be
    - 8-bit
    - 16-bit
    - 32-bit
  - Data transfer cycles can be Single Cycle or Block Transfer
  - Single Cycle – Address is sent with each data transfer
  - Block Transfer – one address is sent with multiple data transfers

VME bus features

- Data Transfer Cycles (almost never used)
  - Single cycles – D8(O), D8(EO), D16, D32 and MD32
  - Block Transfer – BLT, MBLT, A40BLT
- Mixing different address and data widths
  - You can use different address and data widths based on the application
  - Common examples
    - A16/D8(O) – simple IO boards
    - A32/D32 – high performance modules

VME bus features

- Data Transfer Speed

Topology	Bus Cycle	Maximum Speed
VMEbus IEEE-1014	BLT	40 Mbyte/sec
VME64	MBLT	80 Mbyte/sec
VME64x	2eVME	160 Mbyte/sec
VME320	2eSST	320 ~~- 500+~~ Mbyte/sec