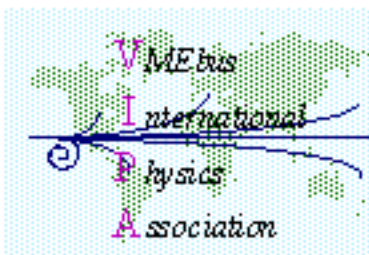


# Chained block transfer (CBLT) and Multi-Cast transfer (MCST).

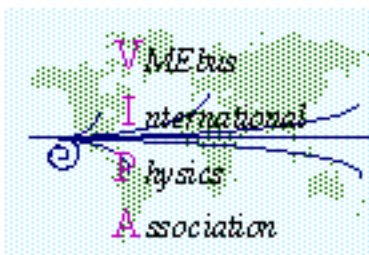
Jan Buytaert

# Compatibility & restrictions.



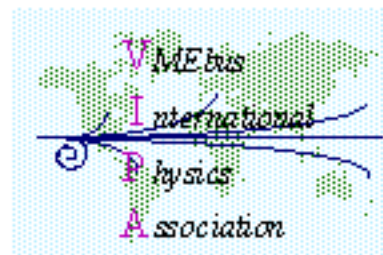
- Fully VME64x compatible !
  - No new AM code.
  - Standard bus masters and backplanes can be used.
  - Only slave interfaces are specific.
- Slaves must have A32.
- Some restriction on module arrangement.
  - No 'VME interrupter' should intervene in between the chained modules.
  - If "active daisy-chain jumpering" backplane, no empty slot allowed inside the chain.
- Appendix E of VME64x.

# CBLT outline



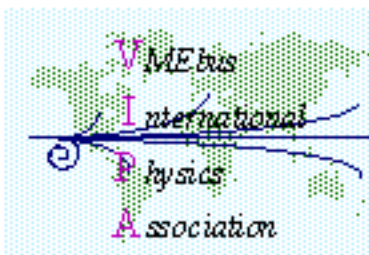
- Sequential readout of multiple slave modules selected by a single address cycle
  - **ordinary** BLT or MBLT read cycle!
  - During **address phase** all slaves in the chain recognize the “key address” and get selected,
  - during **data phase** all slaves respond in turn with available data (token-passing).
  - LAST module **terminates** using BERR\*.

# CBLT details



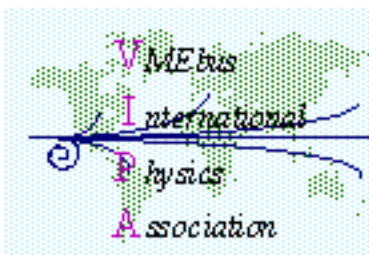
- Key address:
  - [A31...A24]= “chain identification”
  - [A23...A1]= user-defined. (E.g. multi-event-buffer number)
- Chain is statically configured, using CR and CSR space.
  - CBLT capability. (CR-VME64xP)
  - identify FIRST and LAST module (CSR-VME64xP)
  - Enable/disable CBLT (CSR-VME64xP)
  - status register (CR-VME64xP)
  - 'chain identification' register (CSR-VME64xP)

# CBLT details (2)



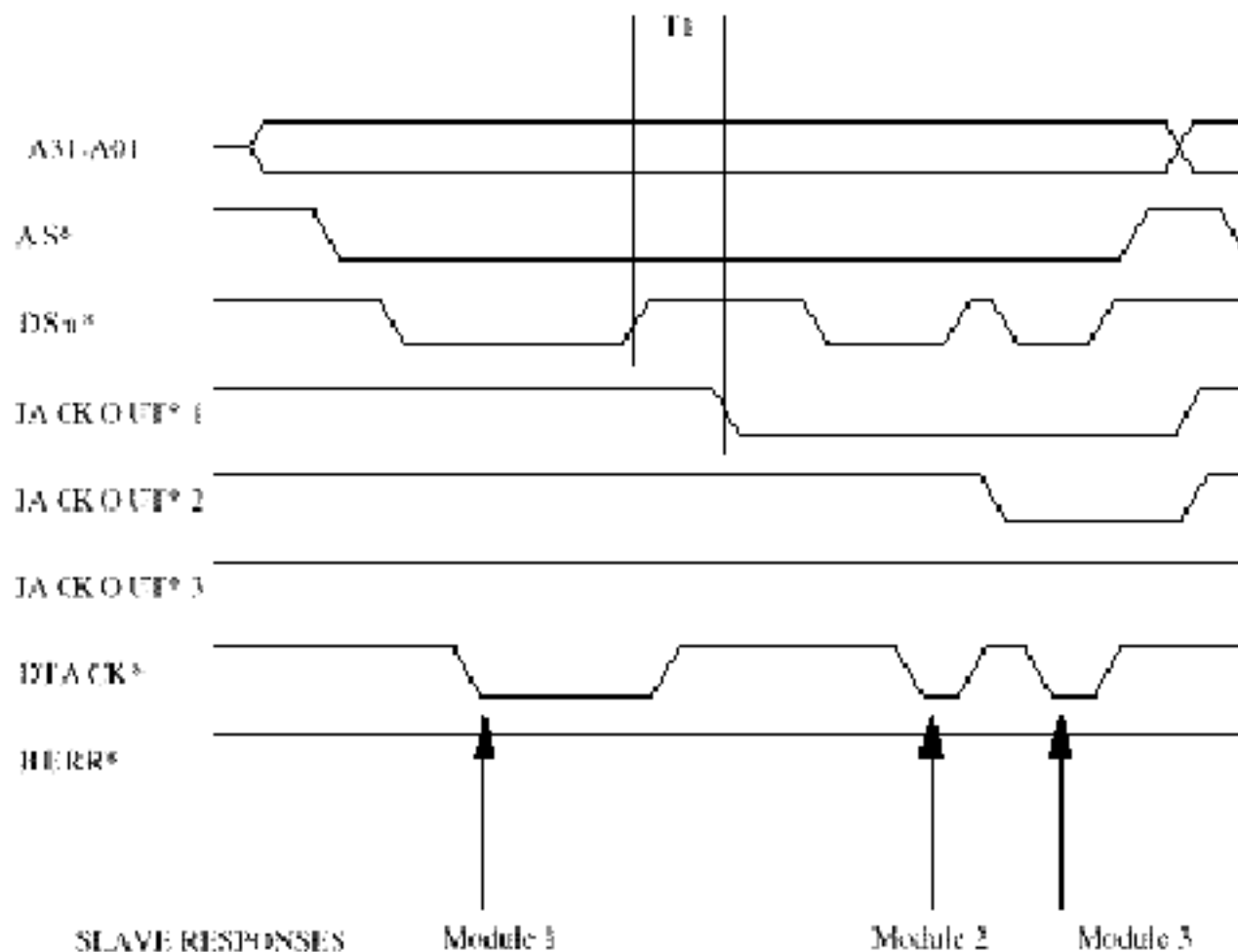
- Token mechanism:
  - Uses IACKin/IACKout daisy chain.
  - Modules obtain token when IACKin\* becomes low.
  - When no more data, slave drives IACKout\* low.
  - FIRST module assumes token immediately ( its IACKin\* not low!)
  - LAST module issues BERR\* (does not propagate IACKout\* low!)
  - EMPTY modules can pass token immediately.

## CBLT details (3)



- Most efficient for **small and sparse** data applications.
- Can handle up to **16M** (BLT) and **4GB** (MBLT) (byte transfer count limit)
- **Multiple** consecutive CBLT's.
- **Data frame** (8 byte) defined.
  - includes transfer byte-count, error/status code, slot position, user-defined bytes.
- Slight complication for MBLT (compelled address phase)

# CBLT timing diagram example



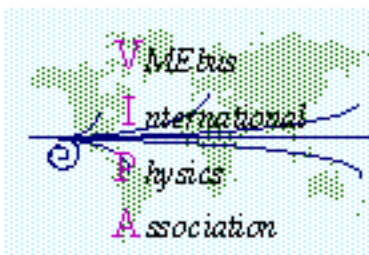
# MCST outline



- Common control of multiple slave modules selected by a single address cycle ;
  - Ordinary single D32 write cycle !
  - During address phase all slaves recognize the 'key-address' and get selected.
  - During data phase all slaves latch the data and pass token.
  - LAST module terminates using DTACK\*.

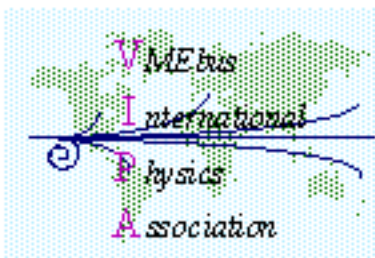


# MCST details



- Key address:
  - [A31...A24]= “chain identification”
  - [A23...A1]= user-defined. (E.g. multi-event-buffer number)
- Chain is statically configured, using CR and CSR space.
  - MCST capability. (CR-VME64xP)
  - identify FIRST and LAST module (CSR-VME64xP)
  - Enable disable MCST (CSR-VME64xP)
  - status register (CR-VME64xP)
  - 'chain identification' register (CSR-VME64xP)

# MCST details (2)



- Token mechanism:
  - Using IACKin/IACKout daisy chain.
  - Modules obtains token when IACKin\* becomes low.
  - When data is latched, slave drives IACKout\* low.
  - FIRST module assumes token immediately ( its IACKin\* not low!)
  - LAST modules issues DTACK\* (does not propagate IACKout\* low!)
- Synchronised control.

# MCST timing diagram

