Panel on System Level Design and VHDL

Peter Ashenden

The University of Adelaide

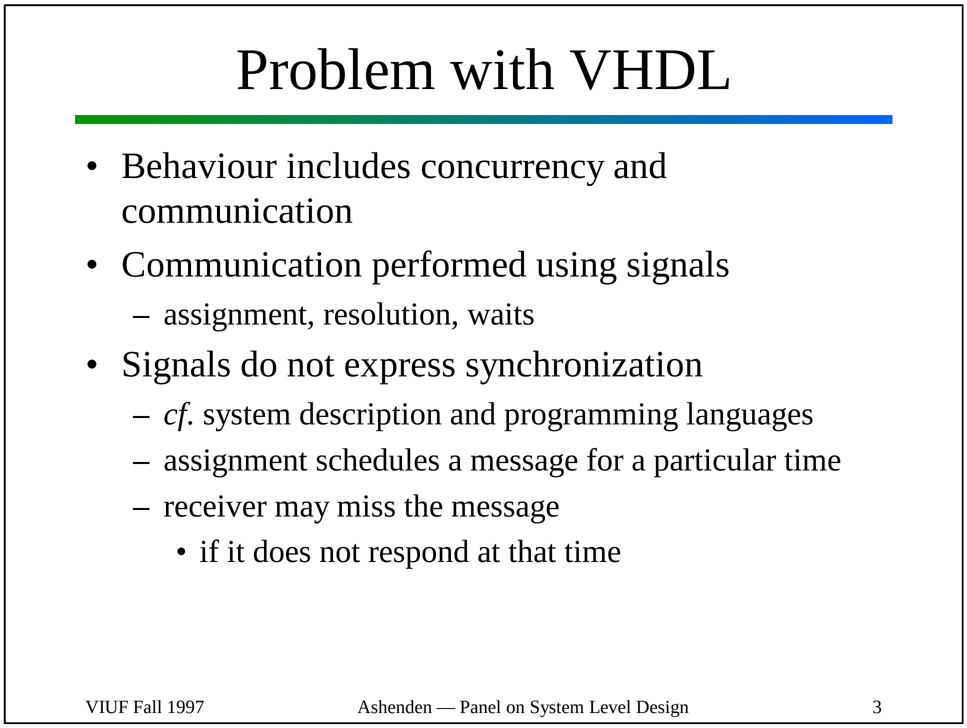
currently Visiting Scholar at

The University of Cincinnati

partially supported by Wright Laboratory
under USAF contract F33615-95-C-1638

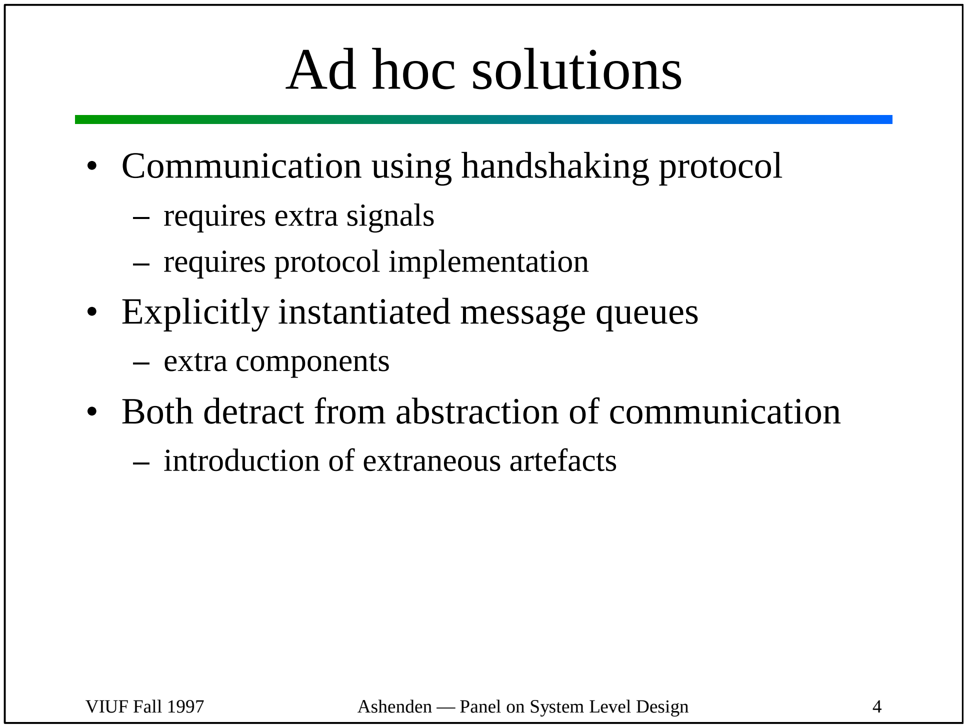
VHDL in the design flow

- At the system level, specify aspects of the system
  - structure (architectural partitioning)
  - behaviour (data transforms, reactions)
  - . . .
- VHDL allows description of structure and behaviour
- Prefer single language approach
  - system level to detailed design (hardware/software)
  - aids refinement
  - avoid semantic mismatch



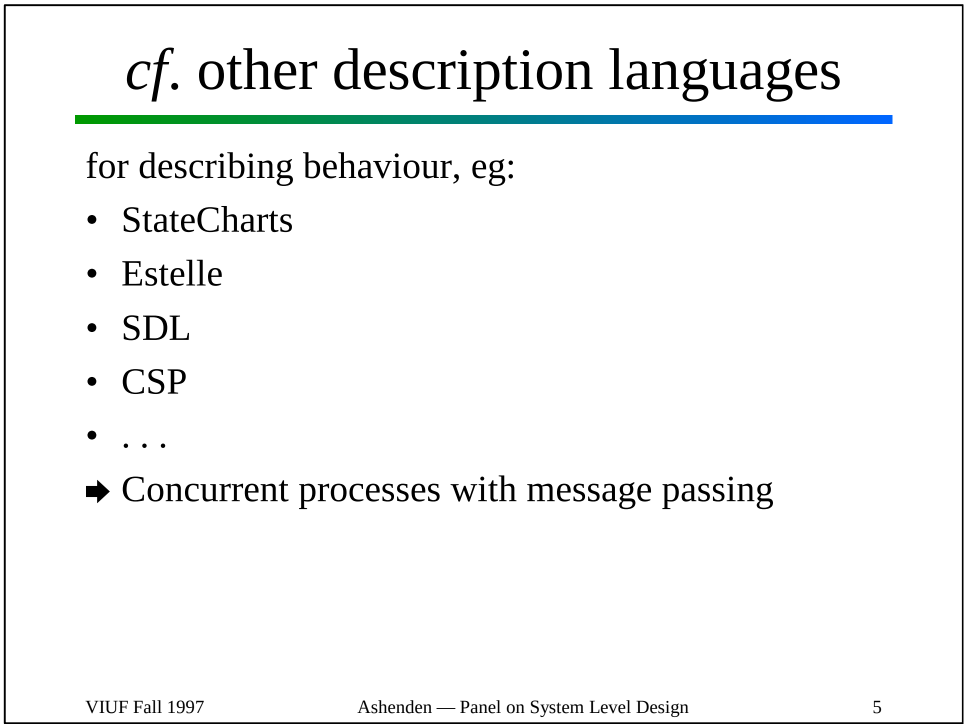
Problem with VHDL

- Behaviour includes concurrency and communication
- Communication performed using signals
  - assignment, resolution, waits
- Signals do not express synchronization
  - *cf.* system description and programming languages
  - assignment schedules a message for a particular time
  - receiver may miss the message
    - if it does not respond at that time



Ad hoc solutions

- Communication using handshaking protocol
  - requires extra signals
  - requires protocol implementation
- Explicitly instantiated message queues
  - extra components
- Both detract from abstraction of communication
  - introduction of extraneous artefacts

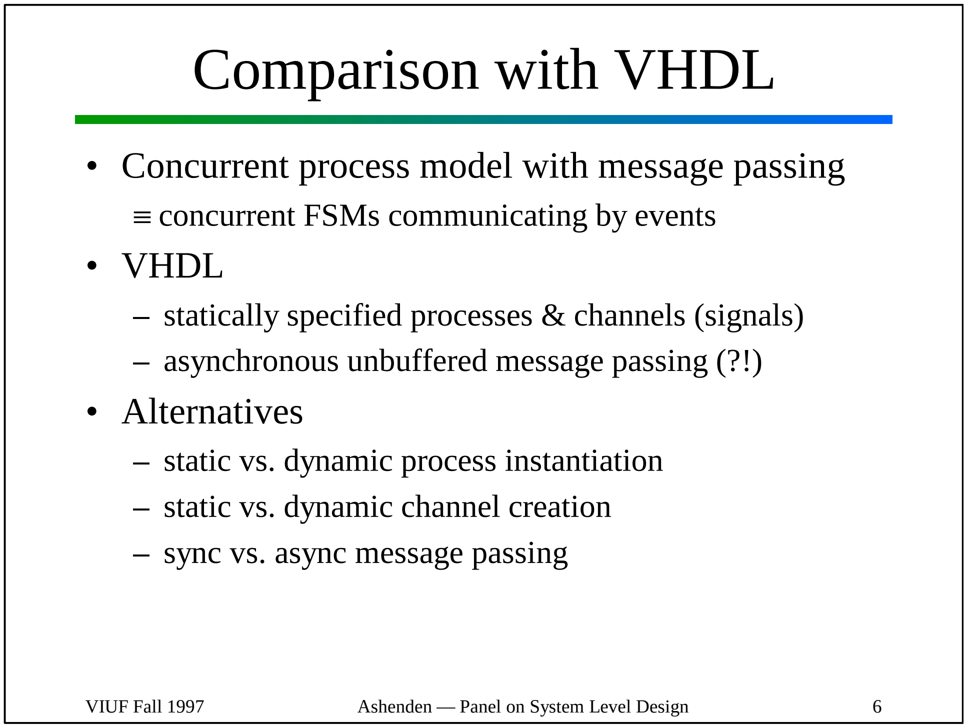


cf. other description languages

for describing behaviour, eg:

- StateCharts
- Estelle
- SDL
- CSP
- . . .

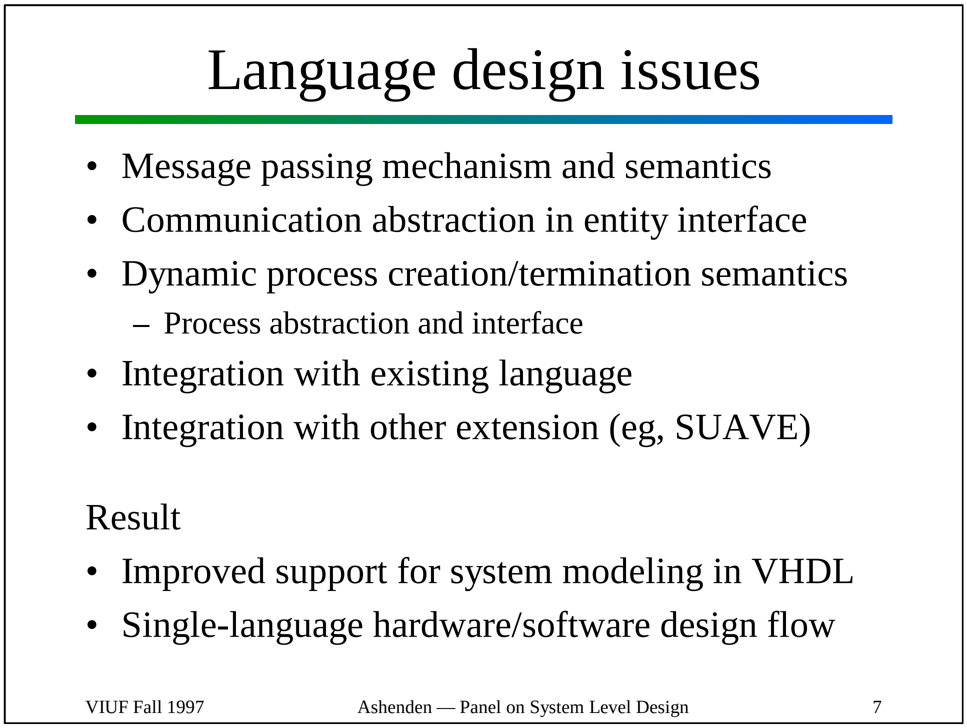
➡ Concurrent processes with message passing



Comparison with VHDL

- Concurrent process model with message passing

  ≡ concurrent FSMs communicating by events
- VHDL
  - statically specified processes & channels (signals)
  - asynchronous unbuffered message passing (?!)
- Alternatives
  - static vs. dynamic process instantiation
  - static vs. dynamic channel creation
  - sync vs. async message passing



Language design issues

- Message passing mechanism and semantics
- Communication abstraction in entity interface
- Dynamic process creation/termination semantics
  - Process abstraction and interface
- Integration with existing language
- Integration with other extension (eg, SUAVE)

Result

- Improved support for system modeling in VHDL
- Single-language hardware/software design flow