Transaction-Level Models for PowerPC and CoreConnect

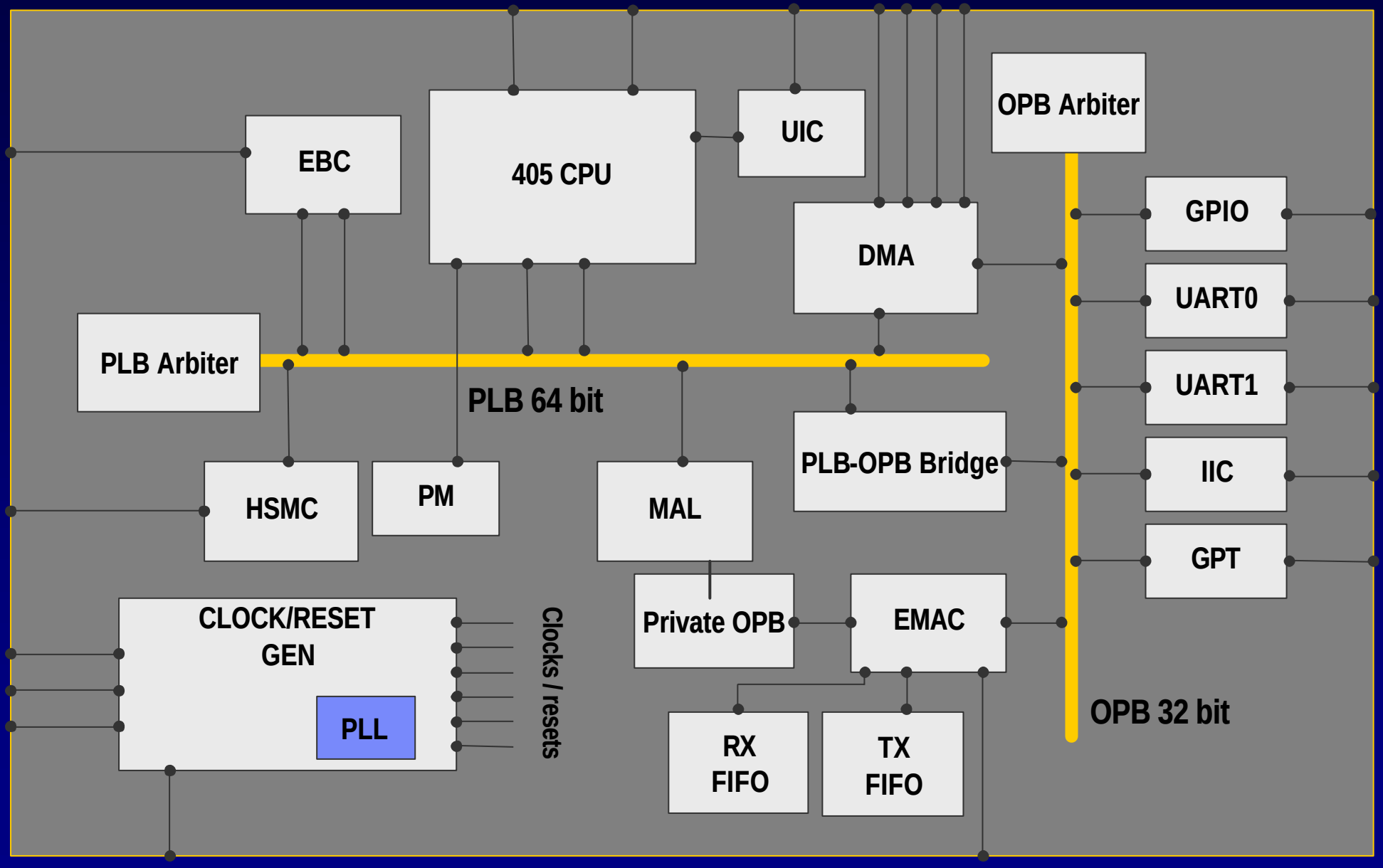
11th European SystemC Users Group Meeting

Reinaldo Bergamaschi
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA

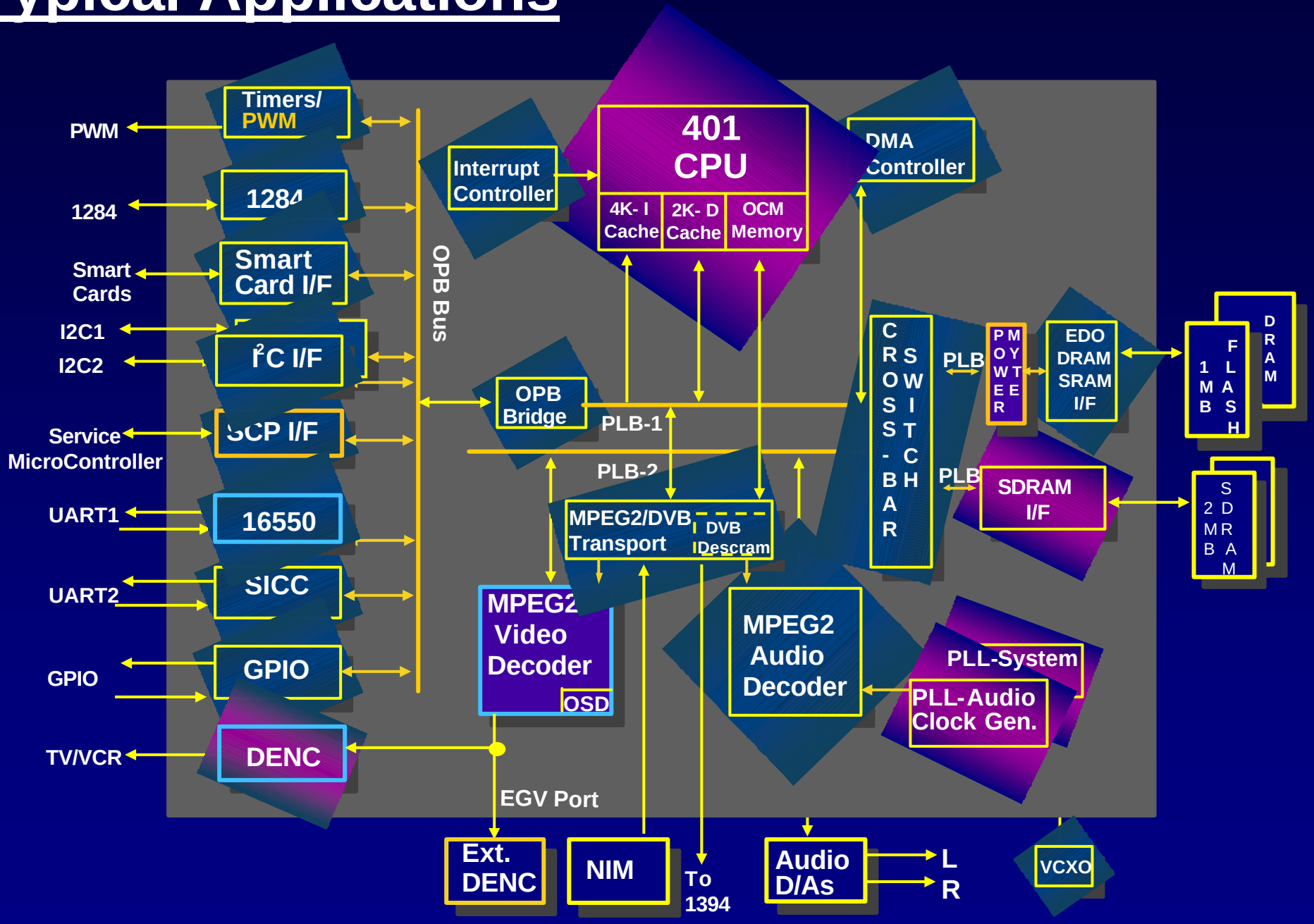
Contents

- SystemC Modeling of IBM's CoreConnect Architecture
- POWER.ORG

Typical Applications



Typical Applications

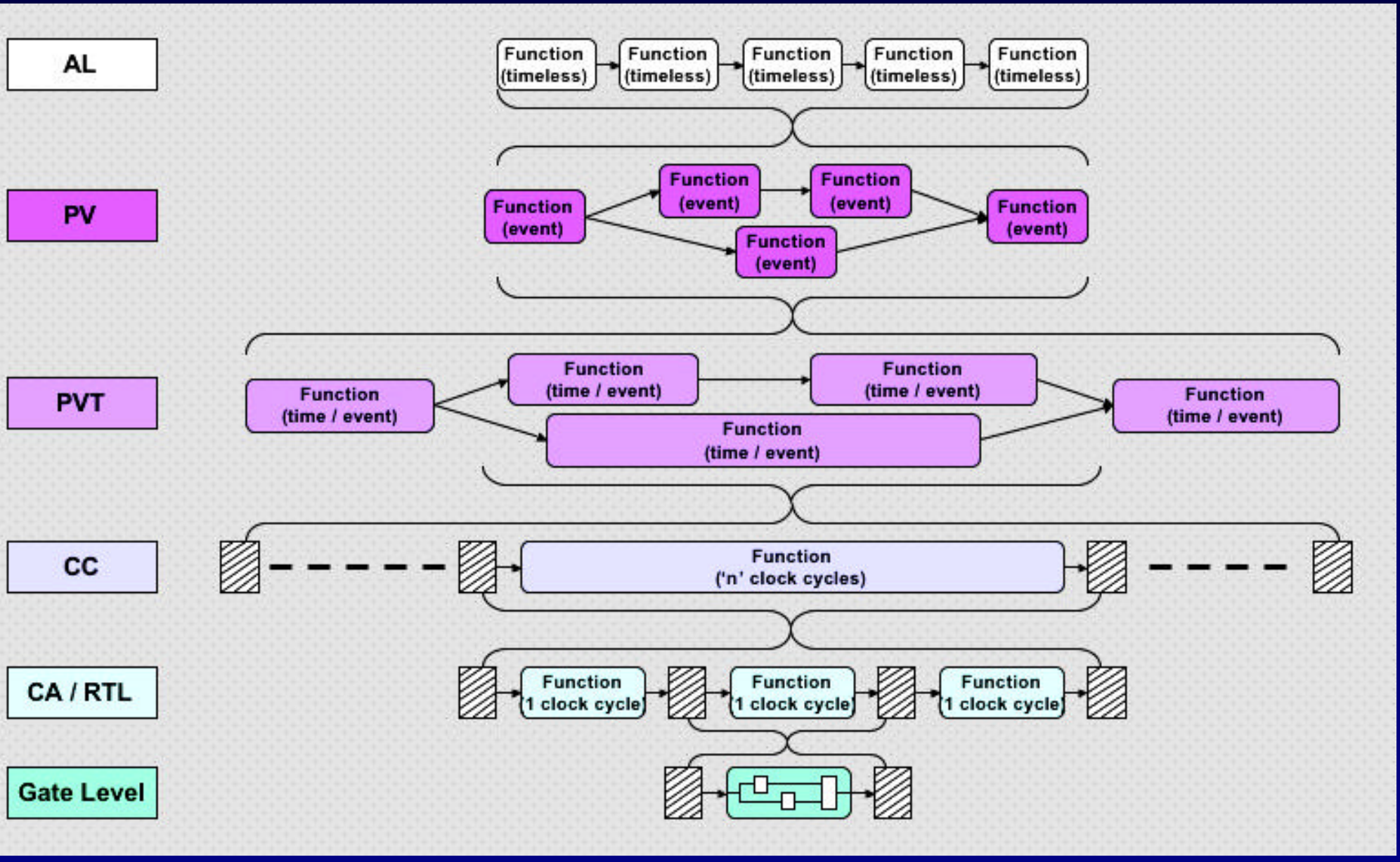


Transaction-Level Modeling

- **Abstractions**
  - **Algorithmic level (AL)**
    - **Architecture/implementation independent, e.g., Matlab, UML**
  - **Programmers View (PV)**
    - **Bit-true representation of the HW, register accurate, no detailed timing**
  - **Programmers View + Timing (PVT)**
    - **Same as PV plus detailed timing and synchronization (Cycle approximate in most cases, accurate in a few cases)**
  - **Cycle Accurate (CA)**
    - **Clocked abstraction, interfaces and transactions**
  - **RTL**
    - **Clocked abstraction, actual chip signals**

- **De-facto implementations**
  - **Mixed of PVT and CA. CA for communication, PVT for computation**

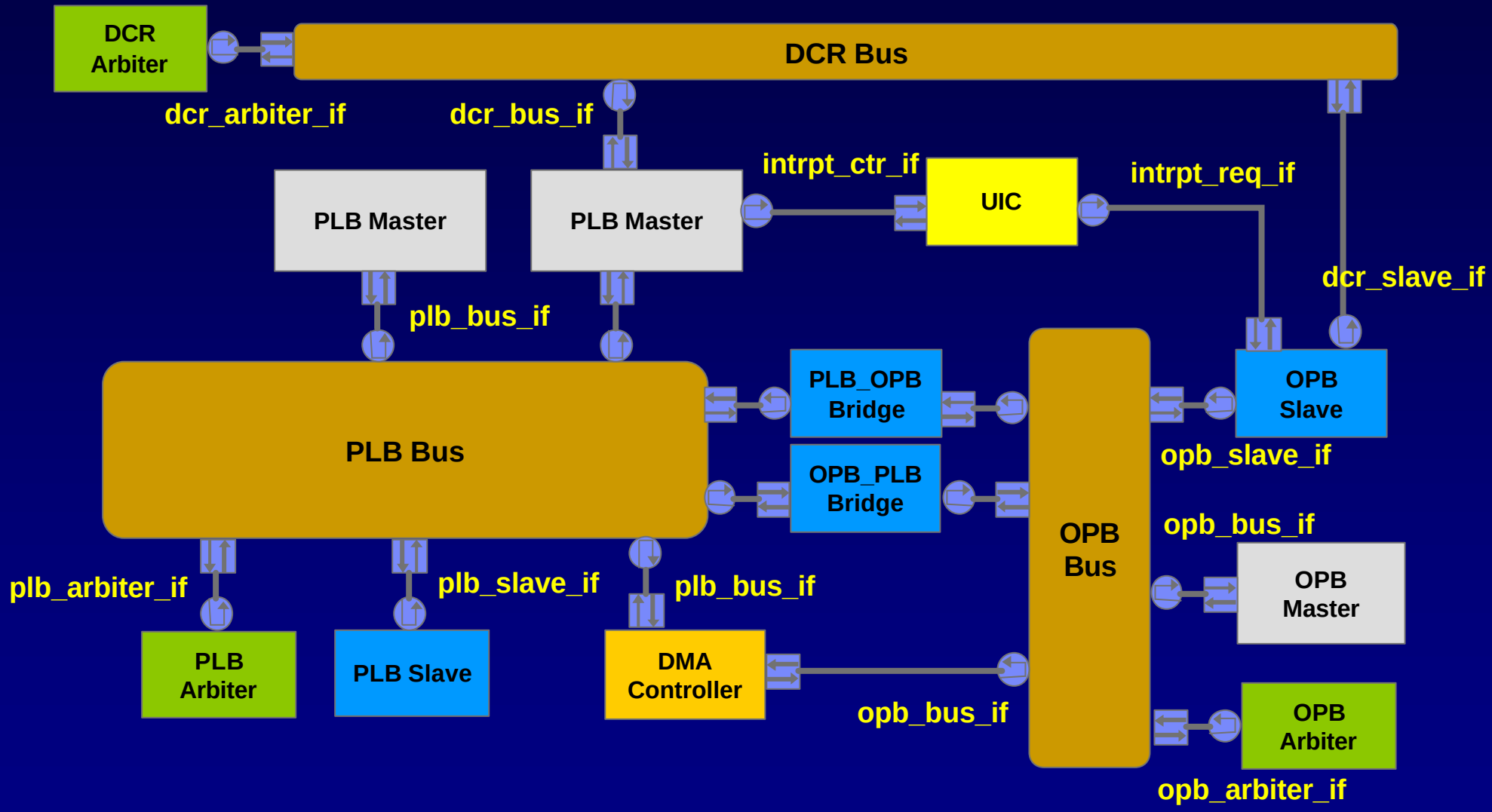
Transaction-Level Modeling Abstractions



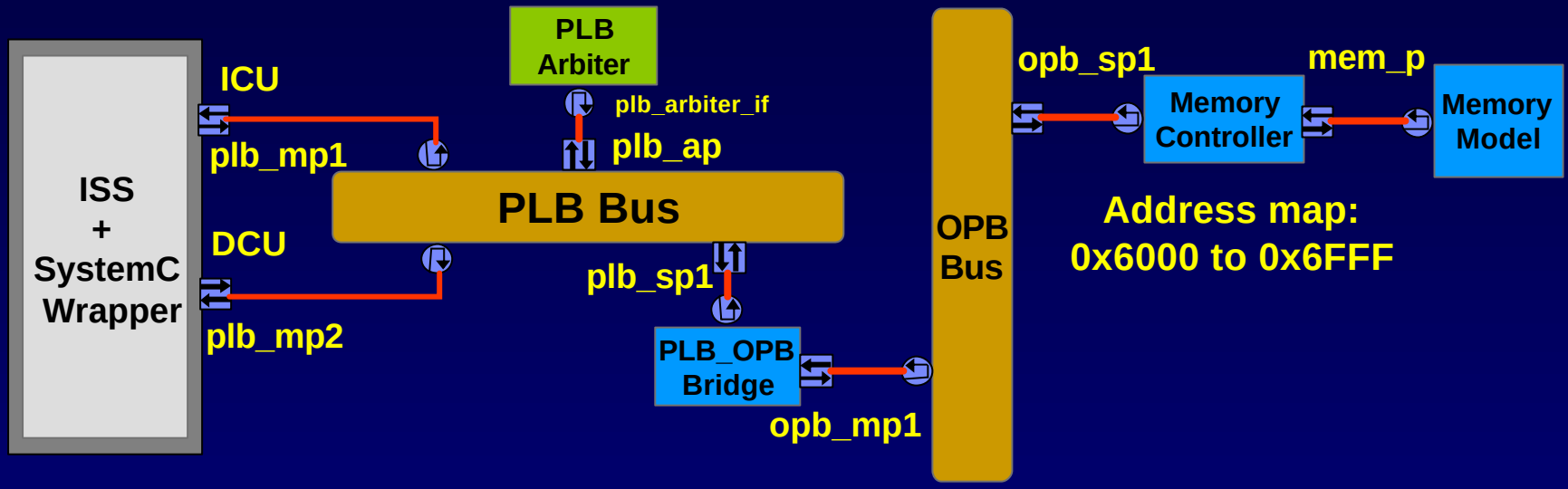
SystemC Modeling of IBM's CoreConnect Architecture

- **Transactions:**
  - High-level functional transactions: CA for communication, PVT for computation.
  - Information is passed via special request structures
  - APIs:
    - blocking_read (plb_request *p), blocking_write (plb_request *p)
    - non_blocking_read (…), non_blocking_write (…)
    - direct_read (…), direct_write (…)
  - All protocol handshaking hidden inside these APIs
  - Users do not need to know how protocol works
- **Goal**
  - Run application code at reasonable speeds (e.g., 100K c/s)
  - Analyze latencies, throughput, bus contention
  - Architectural trade-offs

SystemC Modeling of IBM's CoreConnect Architecture



SystemC Modeling of IBM's CoreConnect Architecture



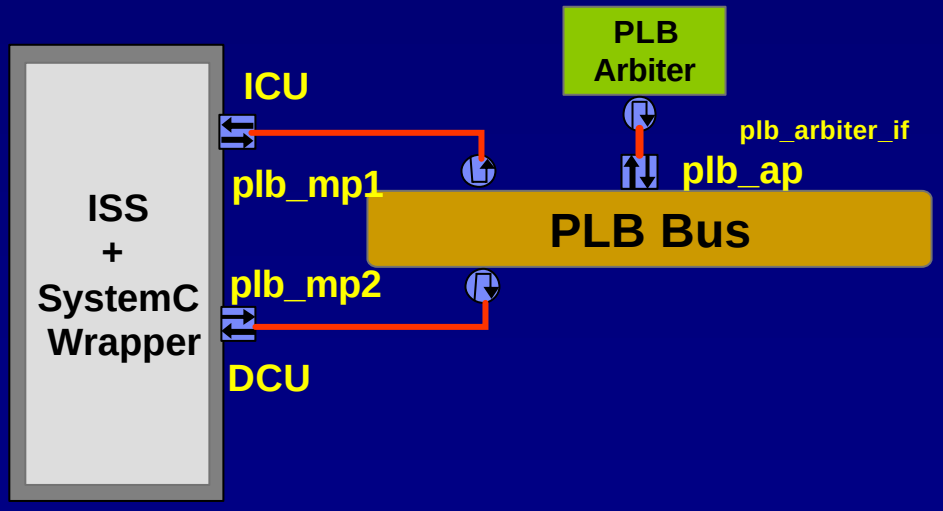

```
#define Ext_Mem 0x6F00

main (..) {
    char *p = Ext_Mem;
    // ...;
    *p = A + B;
    // ....;
}
```

1) Issw.plb_mp2->non_blocking_write (0x6F00, sum(A+B))
2) plb_bus.plb_ap->arbitrate_request(…)
3) Plb queries its slaves and finds plb_opb_bridge mapped to address 0x6F00
4) plb_bus.plb_sp1->write(0x6F00, sum)
5) brg.opb_mp1->non_blocking_write (0x6F00, sum)
6) opb_bus.opb_sp1->write(0x6F00, sum)
7) mc.mem_p->write(0x6F00, sum)
8) Mem[0x6F00] = sum

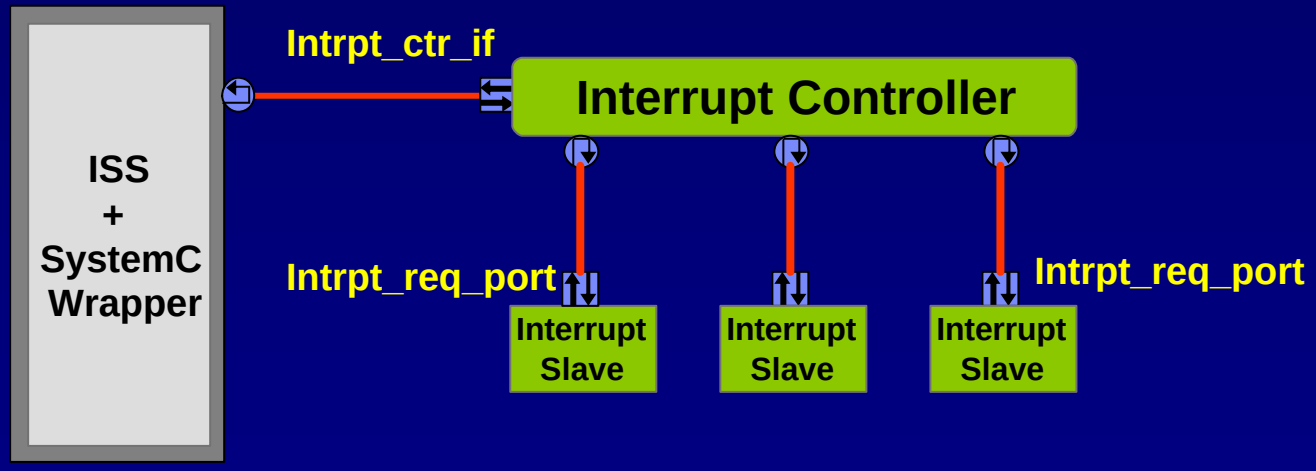
SystemC Modeling of IBM's CoreConnect Architecture

- **Blocking/Non-blocking and Burst/Single transfer**
  - ? These are not the same thing!
  - ? Blocking/Non-blocking have nothing to do with the protocol
  - ? Burst/Single Transfer modes depend on the bus protocol
- **PowerPC 405 can issue a max of 2 PLB requests at a time**
  - ? Instruction fetch (ICU) and Data read/write (DCU)
  - ? Both of them can be burst or single transfer
  - ? But they must be non-blocking (otherwise they can't be issued concurrently)



SystemC Modeling of IBM's CoreConnect Architecture

- **Interrupt Transactions**
  - ? **Interrupt Slave issues interrupt requests through the intrpt_req_port**
  - ? **The Interrupt controller check whether the request is enabled/critical/non-critical, and passes the request to the CPU via the intrpt_ctr_if port**



POWER.ORG

- Power.org is an open developer community and standards organization whose mission is to develop and enable specifications for the promotion and expansion of the Power Architecture solutions and ecosystem

- Members intend to define open Power Architecture specifications initially focusing on:
  - Bus Architecture specifications
  - Server platform specifications
  - Other workgroups may be defined in the future

- SystemC models will be made available to the Power.org community later this year

POWER.ORG

- Initial List of members:
  - AMCC
  - Bull
  - Cadence Design Systems
  - Chartered Semiconductor Manufacturing
  - Culturecom
  - IBM
  - Jabil Circuit
  - Novell
  - Red Hat
  - Sony
  - Shanghai Belling
  - Synopsys
  - Thales
  - Tundra Semiconductor
  - Wistron