OVL Examples

Mike Turpin / ARM

Last Updated: 21st June 2006

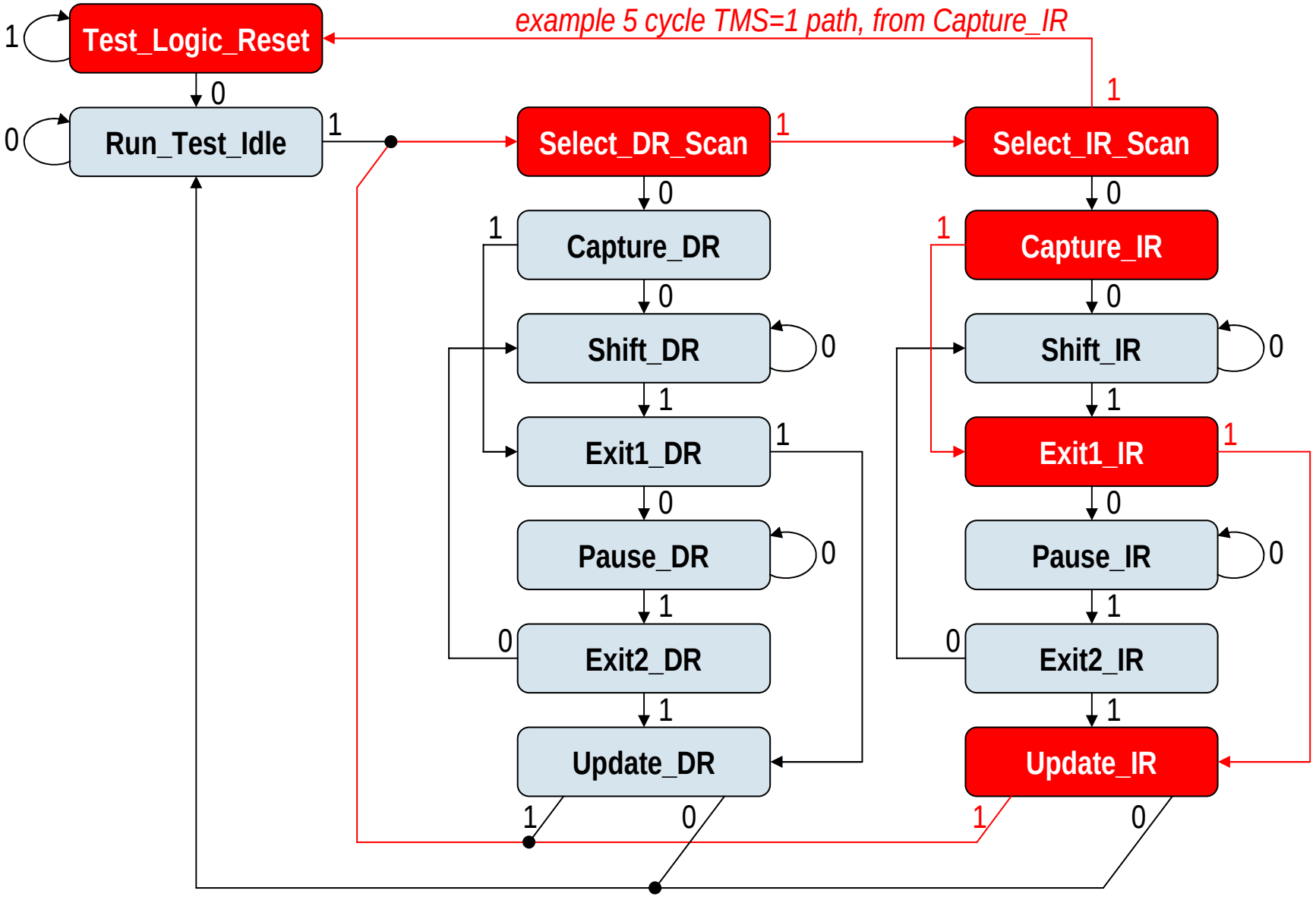
Contents

- Introduction
- TAP Controller FSM
  - State machine example
- Arbiter for Bus #1
  - Reusing OVL to develop a design:
    - round-robin: simple, look-ahead-1, look-ahead-16
    - priority
- Arbiter for Bus #2 *(not yet available)*
  - demonstrator for new ovl_<checker>
    - additional bus_ready input drives new “enable” ports

TAP Controller FSM

- Standard example of a state machine
  - FSM documented in IEEE 1149.1-1990
- IEEE 1149.1 specifies a useful property
  - Whatever starting state, holding the Test Mode Select (TMS) line high for 5 clock cycles will reset the state machine (i.e. next state will be to Test_Logic_Reset)
  - TMS input can be reused as a synchronous reset
- Simple design to introduce OVL
  - OVL from IEEE Specification (described above)
  - OVL describing FSM transition (lower value)

TAP Controller FSM



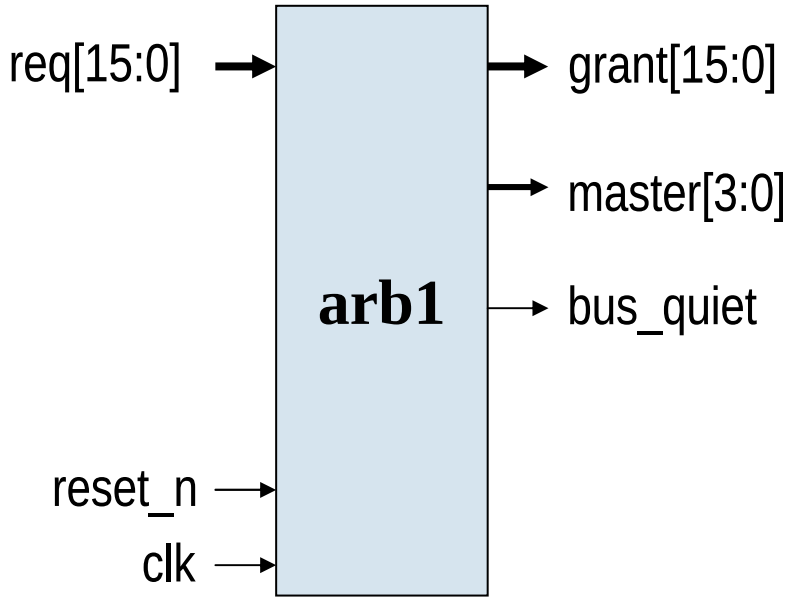
Aside: TAP Controller FSM Encodings

- Avoid FSM encodings with redundant state
  - e.g. onehot
- Otherwise, problem if no async reset:
  - Could power up in unreachable state
  - Could lock up in a set of unreachable states
  - TMS high for 5 cycles would no longer be true
- If no async reset:
  - Use 4-bit binary encoding for the 16 states

Arbiter for Bus #1

- Non-standard bus
  - Created for this OVL example
  - Similar to AHB, but simplified
- 4 arbitration schemes considered
  - Fixed priority
  - Round Robin: simple, look-ahead-1, look-ahead-16
- OVL assertions are common to all designs
  - Shared via `include file
  - Template for standard busses (good reuse)
- Illustrates *Assertion-Based-Design* approach
  - Tradeoffs explored via assertions
  - Round-Robin evolved to improve bus usage

Arbiter for Bus #1: Black Box Specification



- **combinatorial grant**
  - same cycle as request
  - arbiter can delay grant
    - request must be held
- **transmit sequence**
  - start 1 cycle after granted
  - indicated by ~bus_quiet
  - burst length up to 8 cycles
    - request held during burst
- **bus ownership**
  - indicated by master
  - bus_quiet: inactive bus
  - ~bus_quiet: transmitting
- **arbitration not specified**
  - explored via assertions

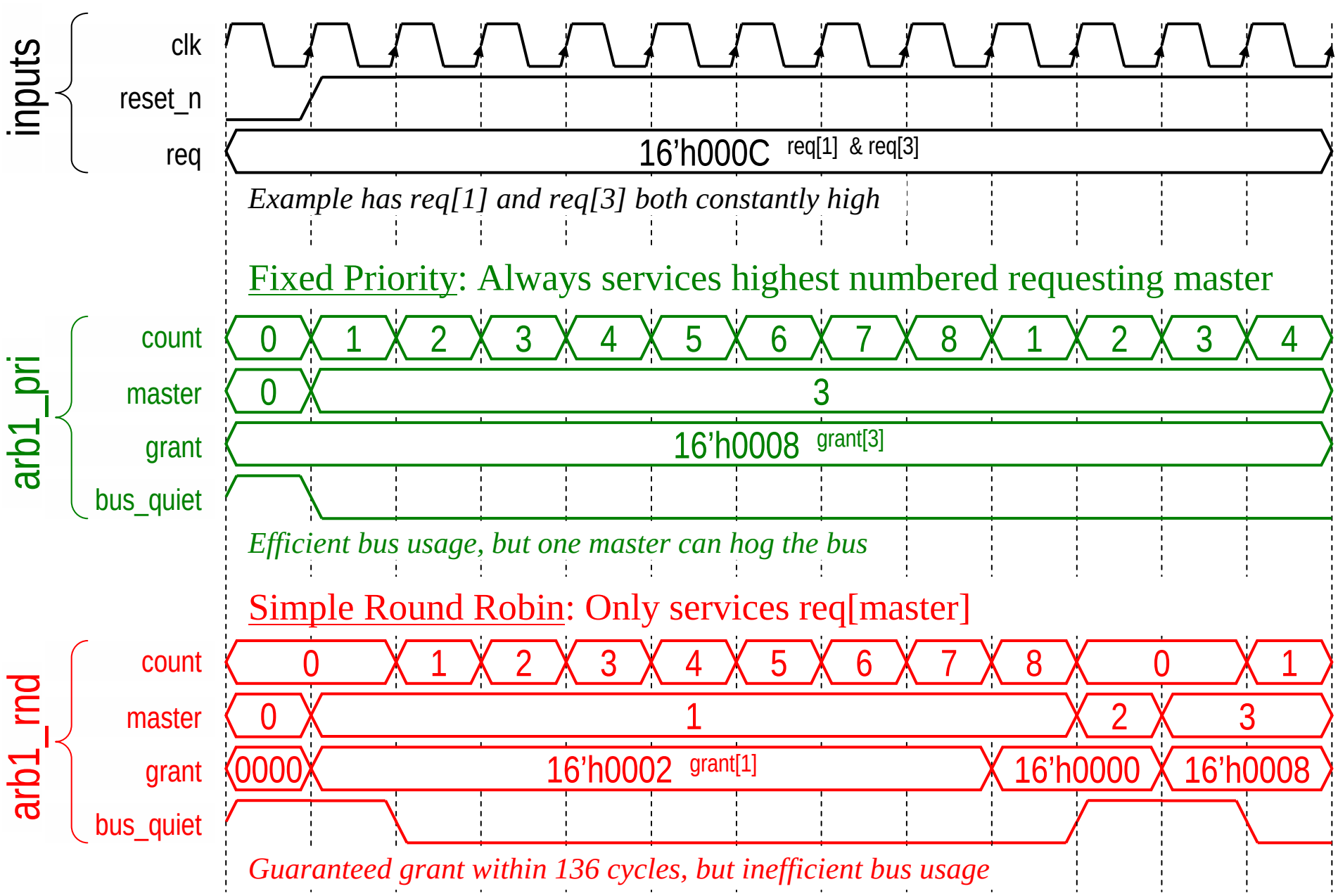
OVL Assertions for Arbiter

- Never more than one grant
  - assert_zero_one_hot
- No unwanted grants (only if requested)
  - assert_implication
- Grant within N cycles
  - assert_frame (16 instances for vlog95, generate for sva31a)
- No hogging the bus
  - assert_change (aux-logic for vlog95, $countones for sva31a)
- Maximum bus usage
  - assert_frame (16 instances for vlog95, generate for sva31a)

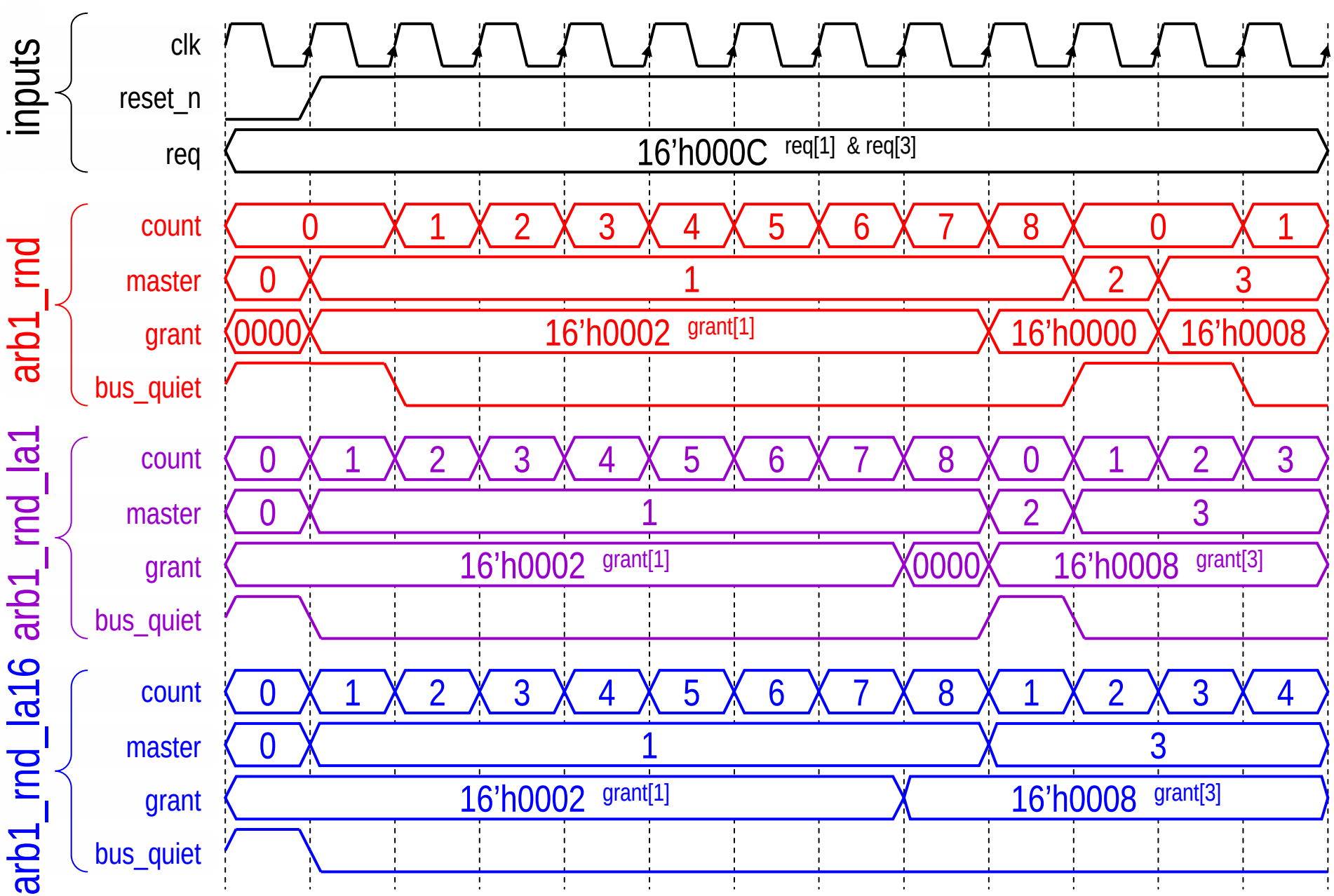
Arbitration Schemes

- Fixed Priority
  - Services request with highest master number
  - Efficient bus usage, but a master can hog the bus
- Simple Round Robin
  - Only services req[master]
  - Guaranteed service within 136 cycles, but inefficient bus usage
- Improved Round Robin Schemes
  - look-ahead-by-1
    - Within 121 cycles (improves 136 for simple round-robin)
  - look-ahead-by-16
    - Within 121 cycles, and efficient bus usage

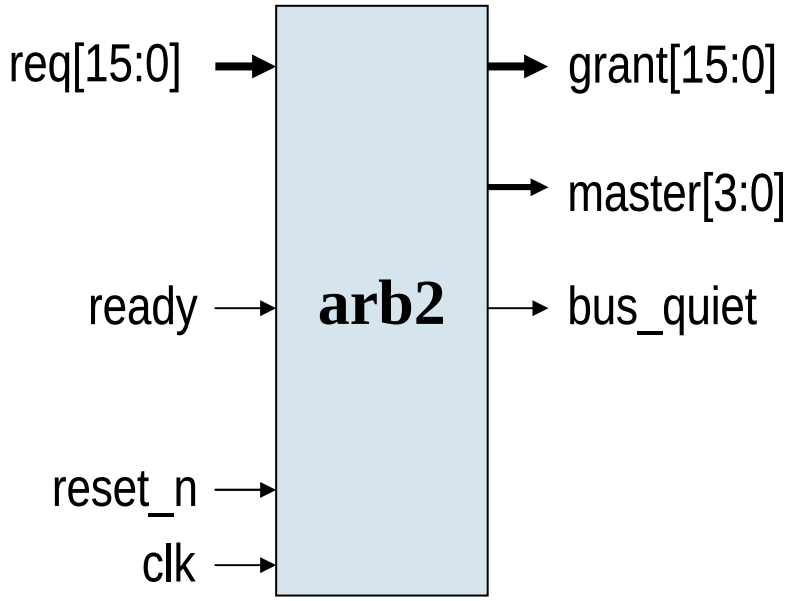
Arbiter #1: Priority vs. Round Robin



Arbiter #1: Look-Ahead improves Round Robin



Arbiter for Bus #2: Black Box Specification



- Arbiter for Bus #2
  - very similar to bus #1
  - closer to AHB
- New: "ready" input
  - response from slaves
  - transmit waits if ~ready
- New ports in OVL v2.?
  - use ovl_<checker>
  - .enable(~ready)

Not yet available, but will be for later versions of OVL