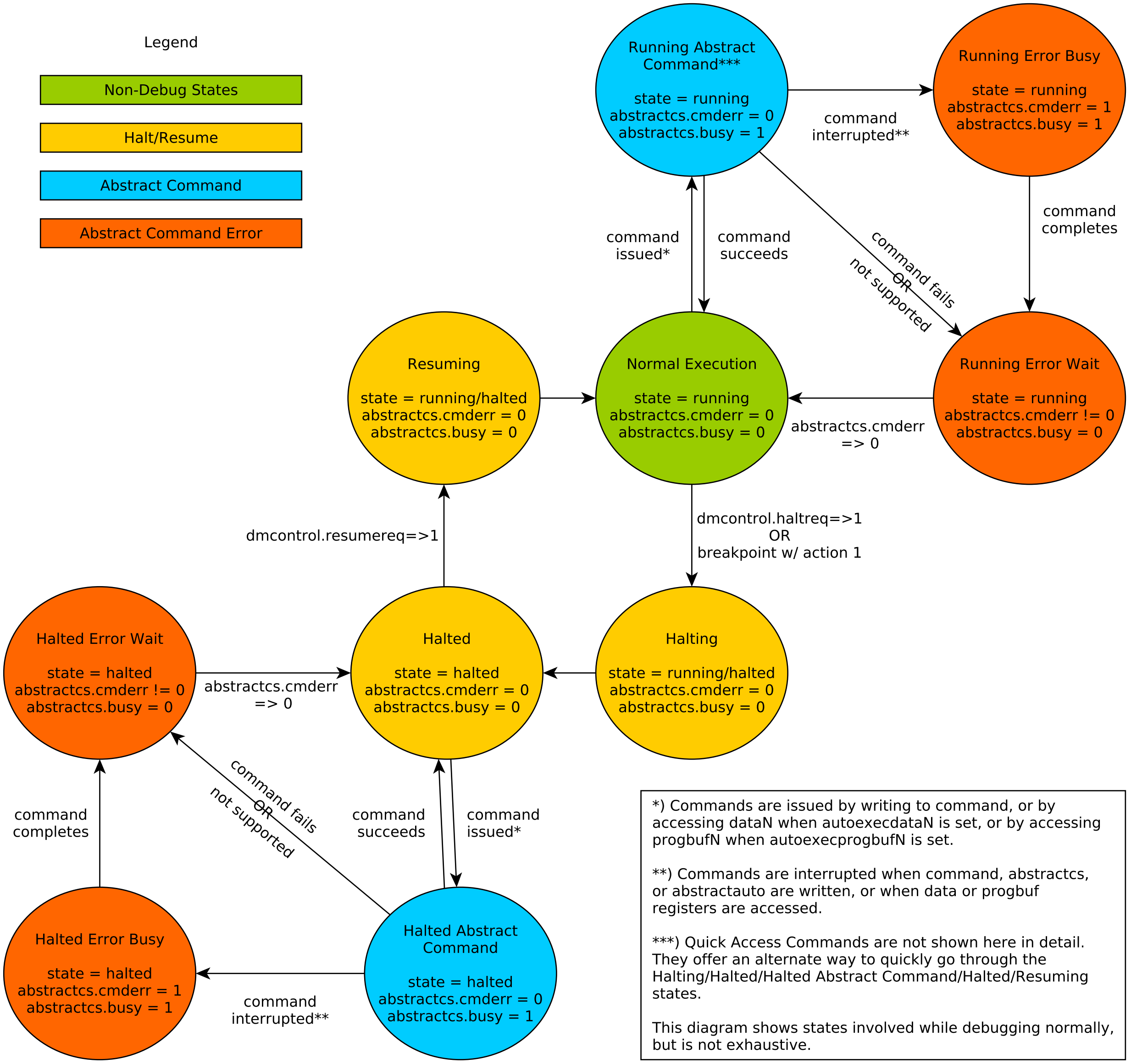
Legend
Non-Debug States
Halt/Resume
Abstract Command
Abstract Command Error
Running Abstract Command***
state = running
abstractcs.cmderr = 0
abstractcs.busy = 1
Running Error Busy
state = running
abstractcs.cmderr = 1
abstractcs.busy = 1
command interrupted**
command issued*
command succeeds
command fails OR not supported
command completes
Resuming
state = running/halted
abstractcs.cmderr = 0
abstractcs.busy = 0
Normal Execution
state = running
abstractcs.cmderr = 0
abstractcs.busy = 0
Running Error Wait
state = running
abstractcs.cmderr != 0
abstractcs.busy = 0
abstractcs.cmderr => 0
dmcontrol.resumereq=>1
dmcontrol.haltreq=>1 OR breakpoint w/ action 1
Halted Error Wait
state = halted
abstractcs.cmderr != 0
abstractcs.busy = 0
abstractcs.cmderr => 0
Halted
state = halted
abstractcs.cmderr = 0
abstractcs.busy = 0
Halting
state = running/halted
abstractcs.cmderr = 0
abstractcs.busy = 0
command completes
command fails OR not supported
command succeeds
command issued*
Halted Error Busy
state = halted
abstractcs.cmderr = 1
abstractcs.busy = 1
command interrupted**
Halted Abstract Command
state = halted
abstractcs.cmderr = 0
abstractcs.busy = 1
*) Commands are issued by writing to command, or by accessing dataN when autoexecdataN is set, or by accessing progbufN when autoexecprogbufN is set.
**) Commands are interrupted when command, abstractcs, or abstractauto are written, or when data or progbuf registers are accessed.
***) Quick Access Commands are not shown here in detail. They offer an alternate way to quickly go through the Halting/Halted/Halted Abstract Command/Halted/Resuming states.
This diagram shows states involved while debugging normally, but is not exhaustive.