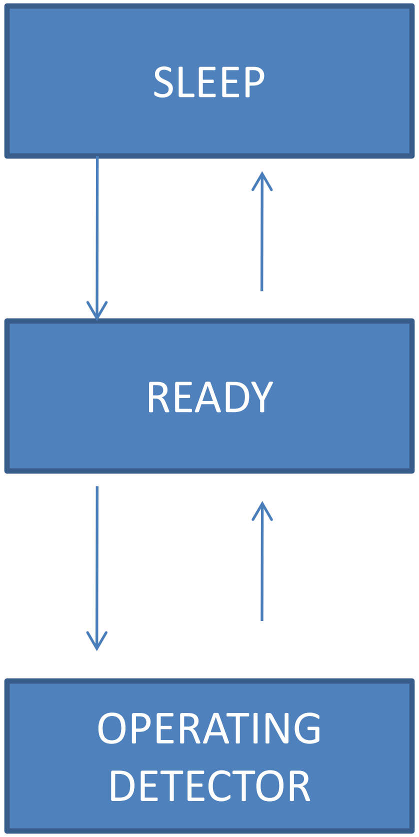
DIF states

CALICE technical meeting on EVO
Thursday, March 5
14:00 CEST
Duration : 1h

DIF states

Motivations for minimal FSM

- States should correspond to HW configurations (state) but not to the running operation (mode)
- Configuration can be done in registers (e.g. put the DIF in debug mode when in READY state)
- High level state would be controlled at high level (e.g. SW)
- Same FSM for everybody
  - Specific functions can be implemented and driven through registers and disabled for everybody when in OP.DET state



=> Agreement within the DIF task force

DIF states

Initial proposal (RC)

SLEEP : links are enabled to receive a wake up command, assume power down by really turning OFF the power at PSU level

READY : DIF can run everything if properly configured (appropriate config. data loaded in registers, commands, ext trig,….)

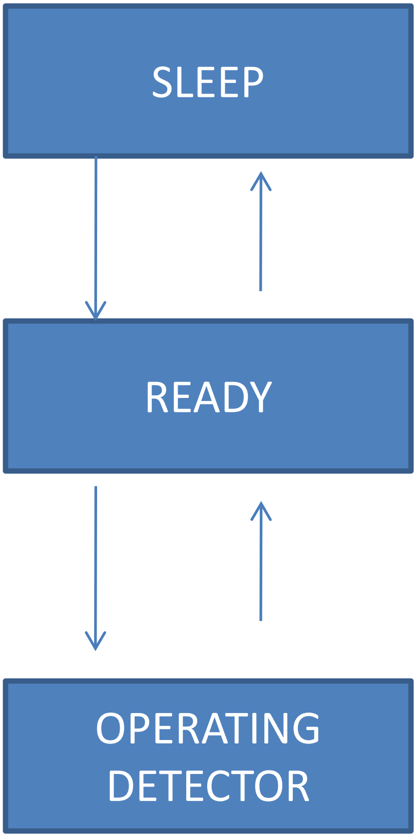
OP. DET. : DIF can only run things that can not disturb the chips, autonomous detector interface manage the chips (no SC, start/stop commands only, read-out by "DMA" to a memory or to the LDA link), HW made functions for triggering etc...

DIF specifications 1.10

SLEEP: Detector is powered-down, DIF still communicates with the DAQ (LDA). DIF configuration (register write and read) is possible. ASICs slow-control configuration can take place in this mode.

READY: All operations are allowed. ASICs slow-control configuration can take place in this mode. Measurements can only be started if the detector is properly configured (slow control data is loaded into ASICs), no errors have occurred.

OPERATING DETECTOR: DIF can only execute commands that do not disturb the current datataking/ data-conversion/readout operation (i.e. no slow-control loading, no debugging). ASICs are operated autonomously by the DIF-ASIC interface. All necessary functions for operation (e.g. trigger) are generated by hardware.



RC’s comments

Some parts of the DIF might be also in low power mode ?

SLEEP: Detector is powered-down, DIF still communicates with the DAQ (LDA). DIF configuration (register write and read) is possible. ~~ASICs slow-control configuration can take place in this mode.~~

READY: All operations are allowed. ASICs slow-control configuration can take place in this mode. Measurements can only be started if the detector is properly configured (slow control data is loaded into ASICs), no errors have occurred.

OPERATING DETECTOR: DIF can only execute commands that do not disturb the current datataking/ data-conversion/readout operation (i.e. no slow-control loading, no debugging). ASICs are operated autonomously by the DIF-ASIC interface. All necessary functions for operation (e.g. trigger) are generated by hardware.

State Transitions

DIF specifications 1.10

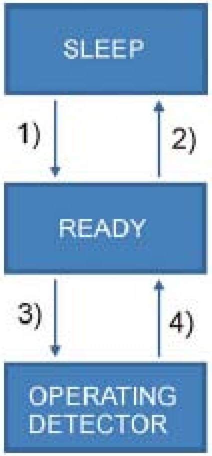
State Transitions: Commands & Conditions (see Fig. 4):

1) Transition executed on: power_on, pre_spill_indication (asynchronous, see Fig. 2), set_DIF_mode, power_pulsing (debugging)

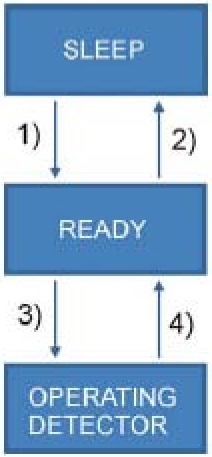
2) Transition executed on: "turn power regulators off" set by option of power_on command, set_DIF_mode, power_pulsing (debugging)

3) this transition is allowed only when slow_control configuration is done and no errors occurred. Transition executed on: start_acquire, pre_spill_indication (synchronous, see Fig. 3), read_results command.

4) Transition executed on: "stop_data taking" set by option of start_acquire command, RAMFull/SCASat condition, "readout_results done" condition, reset command (all options)



RC’s comments



State Transitions: Commands & Conditions (see Fig. 4):

1) Transition executed on: power_on, pre_spill_indication (asynchronous, see Fig. 2), set_DIF_mode, power_pulsing (debugging)

2) Transition executed on: “turn power regulators off” set by option of power_on command, set_DIF_mode, power_pulsing (debugging)

3) this transition is allowed only when slow_control configuration is done and no errors occurred. Transition executed on: start_acquire, pre_spill_indication (synchronous, see Fig. 3), read_results command.

4) Transition executed on: “stop_data taking” set by option of start_acquire command, RAMFull/SCASat condition, “readout_results done” condition, reset command (all options)

End readout signal from last chip ?

Lower level FSM

- For ROCs :

FSM with exclusive states corresponding to the running operation : SControl, ACQuire, CONvert, ROut, …

Transient states for fine sequencing

E.g. POWERUp = PUclk -> PUanalog -> Pudigital etc…

For SC (flash)

For RO packet buiding

Etc….

Are things already defined / used ?