```
machine Mach_PartProc_Trans
//* ************************************************
//    The Event-B model of ARINC 653 Part 1
//    Created by Yongwang Zhao ( zhaoyongwang@gmail.com)
//    National Key Laboratory of Software Development Environment (NLSDE)
//    School of Computer and Engineering, Beihang University, Beijing, China
//    ************************************************/
//this refinement defines the process state transitions and refines the mode transition by adding processes of the partition */
 refines Mach_Part_Trans   sees Ctx_PartProc_Trans

variables processes
          processes_of_partition // system_has_inited
          partition_mode process_state

invariants
  @inv_proc processes ∈ ℙ(PROCESSES)
  /* created processes will not exceed 1024
     @inv_part_mode partition_mode ∈ PARTITIONS → PARTITION_MODES */
  @inv_proc_state process_state ∈ processes → PROCESS_STATES
  @inv_proc_of_part processes_of_partition ∈ processes → PARTITIONS // total function
```

@inv_readyrunsuspproc_onlyin_normalpart ∀*p*·(*p*∈**PARTITIONS** ∧ partition_mode(*p*) ≠ **PM_NORMAL** ⇒ ∀proc·(proc ∈ processes_of_partition~[{*p*}] ⇒ process_state(proc) ≠**PS_Ready** ∧ process_state(proc) ≠ **PS_Running** ∧ process_state(proc) ≠**PS_Suspend**)) *// the process will not in ready or running state, if its partition is not in NORMAL*

@inv_readyrunsusp_proc_imply_normalpart ∀*proc*·(*proc* ∈ processes ∧ (process_state(*proc*)=**PS_Ready** ∨ process_state(*proc*)=**PS_Running** ∨ process_state(*proc*)=**PS_Suspend**) ⇒ partition_mode(processes_of_partition(*proc*)) = **PM_NORMAL**) *// a process is in ready or run state, its partition must in normal mode*

@inv_noproc_imply_notnormal ∀*part*·(*part* ∈ **PARTITIONS** ∧ *part*∈ran(processes_of_partition) ∧ card(processes_of_partition~[{*part*}])=0 ⇒ partition_mode(*part*) ≠ **PM_NORMAL**) *// if there is not process in one partition, it should not be started (in NORMAL mode)*

@inv_normalmode_imply_procs ∀*part*·(*part* ∈ **PARTITIONS** ∧ partition_mode(*part*) = **PM_NORMAL** ⇒ *part*∈ ran(processes_of_partition) ∧card(processes_of_partition~[{*part*}])> 0) *// the NORMAL partition must have processes*

@inv_idlemode_imply_noproc ∀*part*·(*part* ∈ **PARTITIONS** ∧ partition_mode(*part*) = **PM_IDLE** ⇒ *part*∉ ran(processes_of_partition)) *// the IDLE partition has no process*

@inv_part_mode partition_mode ∈ **PARTITIONS** → **PARTITION_MODES** *// @inv_part partv∈PARTITIONS*

events

event INITIALISATION

```
    then
      @act01 partition_mode :| partition_mode'∈PARTITIONS → {PM_COLD_START,PM_WARM_START}
      /* @act01 partition_mode ≔ PARTITIONS × {PM_COLD_START}
         @act01 partition_mode :| ∀p·(p∈PARTITIONS⇒partition_mode'(p)∈
{PM_COLD_START,PM_WARM_START}) */
      @act00 processes ≔ ∅
      @act02 process_state ≔ ∅
      @act03 processes_of_partition ≔ ∅ // @act04 system_has_inited ≔ FALSE
  end

  event process_schedule
    any part proc
    where
      @grd01 part ∈ PARTITIONS
      @grd02 proc ∈ processes
      @grd03 processes_of_partition(proc) = part
      @grd04 partition_mode(part) = PM_NORMAL
      @grd05 process_state(proc) = PS_Ready ∨ process_state(proc) = PS_Running
    then
      @act1 process_state ≔ (process_state  (process_state~[{PS_Running}] × {PS_Ready}))  {proc ↦
PS_Running}
```

```
  end

  event create_process
    any part proc
    where
      @grd01 part ∈ PARTITIONS
      @grd02 proc ∈ PROCESSES ∖ processes
      @grd03 partition_mode(part)=PM_COLD_START ∨ partition_mode(part)=PM_WARM_START // process
can only be created, when the partition is being initialize
    then
      @act01 processes ≔ processes ∪ {proc}
      @act02 processes_of_partition(proc) ≔ part
      @act03 process_state(proc) ≔ PS_Dormant
  end

  event partition_modetransition_to_idle refines partition_mode_transition
    any part newm procs
    where
      @grd01 part ∈ PARTITIONS
      @grd02 newm ∈ PARTITION_MODES
      @grd03 partition_mode(part) = PM_COLD_START ∨ partition_mode(part) = PM_WARM_START ∨
```

```
partition_mode(part) = PM_NORMAL
        @grd04 newm = PM_IDLE
        @grd07 procs = processes_of_partition~[{part}]
      then
        @act01 partition_mode(part) ≔ newm
        @act22 processes ≔ processes ∖ procs
        @act23 process_state ≔ procs ⩤ process_state
        @act24 processes_of_partition ≔ procs ⩤ processes_of_partition
    end

    event partition_modetransition_to_normal refines partition_mode_transition
      any part newm procs procsstate
          procs2 // procs: some WAITING procs need to be transited to READY, some WAITING procs retain the
state. procs2: WAIT_SUSPEND procs transit to SUSPEND

      where
        @grd01 part ∈ PARTITIONS
        @grd02 newm ∈ PARTITION_MODES
        @grd03 partition_mode(part) = PM_COLD_START ∨ partition_mode(part) = PM_WARM_START
        @grd04 newm = PM_NORMAL
        @grd08 card(processes_of_partition~[{part}]) > 0 // the partition should not goto normal, if there is not
```

process in it.

@grd09 *procs* = processes_of_partition~[{*part*}] ∩ process_state~[{**PS_Waiting**}] *// transit to normal, some WAITING procs (aperiod, not suspended) will be transit to READY*

@grd10 *procs2* = processes_of_partition~[{*part*}] ∩ process_state~[{**PS_WaitandSuspend**}] *// transit to normal, the WAITandSuspend procs will be transit to suspend*

@grd101 *procsstate* ∈ *procs* → {**PS_Waiting**,**PS_Ready**}

then

@act01 partition_mode(*part*) ≔ *newm*

@act22 process_state ≔ (process_state *procsstate*) (*procs2* ×{**PS_Suspend**})

end

event partition_modetransition_to_coldstart *// cold_start or normal transit to cold_start, the processes of this partition should be deleted*

refines partition_mode_transition

any *part newm procs*

where

@grd01 *part* ∈ **PARTITIONS**

@grd02 *newm* ∈ **PARTITION_MODES**

@grd04 *newm* = **PM_COLD_START**

@grd03 partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START** ∨ partition_mode(*part*) = **PM_NORMAL**

```
    @grd08 procs = processes_of_partition~[{part}]
  then
    @act01 partition_mode(part) ≔ newm
    @act22 processes ≔ processes ∖ procs
    @act23 process_state ≔ procs ⩤ process_state
    @act24 processes_of_partition ≔ procs ⩤ processes_of_partition
end

event partition_modetransition_to_warmstart // normal transit to warm_start, the processes of this partition
should be deleted
refines partition_mode_transition
  any part newm procs
  where
    @grd01 part ∈ PARTITIONS
    @grd02 newm ∈ PARTITION_MODES
    @grd04 newm = PM_WARM_START
    @grd09 partition_mode(part) = PM_WARM_START ∨ partition_mode(part) = PM_NORMAL
    @grd08 procs = processes_of_partition~[{part}]
  then
    @act01 partition_mode(part) ≔ newm
    @act22 processes ≔ processes ∖ procs
```

```
    @act23 process_state ≔ procs ⩤ process_state
    @act24 processes_of_partition ≔ procs ⩤ processes_of_partition
end

event partition_modetransition_idle_to_warmstart // idle transit to cold_start or warm_start
refines partition_mode_transition
  any part newm
  where
    @grd01 part ∈ PARTITIONS
    @grd02 newm ∈ PARTITION_MODES
    @grd04 newm = PM_WARM_START
    @grd07 partition_mode(part) = PM_IDLE
  then
    @act01 partition_mode(part) ≔ newm
end

event partition_modetransition_idle_to_coldstart // idle transit to cold_start or warm_start
refines partition_mode_transition
  any part newm
  where
    @grd01 part ∈ PARTITIONS
```

@grd02 *newm* ∈ **PARTITION_MODES**
@grd04 *newm* = **PM_COLD_START**
@grd07 partition_mode(*part*) = **PM_IDLE**
then
@act01 partition_mode(*part*) ≔ *newm*
end

event process_state_transition *// READY --> RUNNING and RUNNING --> READY is in schedule*
any *part proc newstate*
where
@grd01 *part* ∈ **PARTITIONS**
@grd02 *proc* ∈ processes
@grd03 *newstate* ∈ **PROCESS_STATES**
@grd06 processes_of_partition(*proc*) = *part*
@grd07 partition_mode(*part*) ≠ **PM_IDLE**
@grd20 ((partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START**) ∧ process_state(*proc*) = **PS_Dormant**) ⇒ *newstate* = **PS_Waiting**
@grd21 ((partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START**) ∧ process_state(*proc*) = **PS_Waiting**) ⇒ (*newstate* = **PS_Dormant** ∨ *newstate* = **PS_WaitandSuspend**)
@grd29 ((partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START**) ∧ process_state(*proc*) = **PS_WaitandSuspend**) ⇒ (*newstate* = **PS_Dormant** ∨ *newstate* = **PS_Waiting**)

```
        /* this is the correct transition
            the next line is the ARINC653 defined transition, from WAIT --> WAIT without the RESUME action
            @grd29 ((partition_mode(part) = PM_COLD_START ∨ partition_mode(part) = PM_WARM_START) ∧
process_state(proc) = PS_WaitandSuspend) ⇒ (newstate = PS_Dormant) */
        @grd22 (partition_mode(part) = PM_NORMAL ∧ process_state(proc) = PS_Dormant) ⇒ (newstate =
PS_Ready ∨ newstate = PS_Waiting)
        @grd23 (partition_mode(part) = PM_NORMAL ∧ process_state(proc) = PS_Ready) ⇒ (newstate =
PS_Dormant ∨ newstate = PS_Suspend)
        @grd24 (partition_mode(part) = PM_NORMAL ∧ process_state(proc) = PS_Waiting) ⇒ (newstate =
PS_Dormant ∨ newstate = PS_WaitandSuspend ∨ newstate = PS_Ready)
        @grd25 (partition_mode(part) = PM_NORMAL ∧ process_state(proc) = PS_Suspend) ⇒ (newstate =
PS_Dormant ∨ newstate = PS_Ready)
        @grd28 (partition_mode(part) = PM_NORMAL ∧ process_state(proc) = PS_WaitandSuspend) ⇒
(newstate = PS_Waiting ∨ newstate = PS_Suspend ∨ newstate = PS_Dormant)
        @grd27 (partition_mode(part) = PM_NORMAL ∧ process_state(proc) = PS_Running) ⇒  (newstate =
PS_Running ∨ newstate = PS_Ready ∨ newstate = PS_Waiting ∨ newstate = PS_Suspend ∨ newstate =
PS_Dormant)
    then
        @act01 process_state(proc) ≔ newstate
    end
```

event process_state_transition2 *// READY --> RUNNING and RUNNING --> READY is in schedule*

any *part procs newstates*

where

@grd01 *part* ∈ **PARTITIONS**

@grd02 *procs* ⊆ processes

@grd03 *newstates* ∈ *procs* → **PROCESS_STATES**

@grd06 *procs* ⊆ processes_of_partition~[{*part*}]

@grd07 partition_mode(*part*) ≠ **PM_IDLE**

@grd20 ∀*proc*·((*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START**) ∧ process_state(*proc*) = **PS_Dormant**) ⇒ *newstates*(*proc*) = **PS_Waiting**)

@grd21 ∀*proc*·((*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START**) ∧ process_state(*proc*) = **PS_Waiting**) ⇒ (*newstates*(*proc*) = **PS_Dormant** ∨ *newstates*(*proc*) = **PS_WaitandSuspend**))

@grd29 ∀*proc*·((*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_COLD_START** ∨ partition_mode(*part*) = **PM_WARM_START**) ∧ process_state(*proc*) = **PS_WaitandSuspend**) ⇒ (*newstates*(*proc*) = **PS_Dormant** ∨ *newstates*(*proc*) = **PS_Waiting**))

/ this is the correct transition*

the next line is the ARINC653 defined transition, from WAIT --> WAIT without the RESUME action

*@grd29 ((partition_mode(part) = PM_COLD_START ∨ partition_mode(part) = PM_WARM_START) ∧ process_state(proc) = PS_WaitandSuspend) ⇒ (newstate = PS_Dormant) */*

@grd22 ∀*proc*·(*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_NORMAL** ∧ process_state(*proc*) =

PS_Dormant) ⇒ (*newstates*(*proc*) = **PS_Ready** ∨ *newstates*(*proc*) = **PS_Waiting**))

@grd23 ∀*proc*·(*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_NORMAL** ∧ process_state(*proc*) = **PS_Ready**) ⇒ (*newstates*(*proc*) = **PS_Dormant** ∨ *newstates*(*proc*) = **PS_Suspend**))

@grd24 ∀*proc*·(*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_NORMAL** ∧ process_state(*proc*) = **PS_Waiting**) ⇒ (*newstates*(*proc*) = **PS_Dormant** ∨ *newstates*(*proc*) = **PS_WaitandSuspend** ∨ *newstates*(*proc*) = **PS_Ready**))

@grd25 ∀*proc*·(*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_NORMAL** ∧ process_state(*proc*) = **PS_Suspend**) ⇒ (*newstates*(*proc*) = **PS_Dormant** ∨ *newstates*(*proc*) = **PS_Ready**))

@grd28 ∀*proc*·(*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_NORMAL** ∧ process_state(*proc*) = **PS_WaitandSuspend**) ⇒ (*newstates*(*proc*) = **PS_Waiting** ∨ *newstates*(*proc*) = **PS_Suspend** ∨ *newstates*(*proc*) = **PS_Dormant**))

@grd27 ∀*proc*·(*proc*∈*procs* ∧ (partition_mode(*part*) = **PM_NORMAL** ∧ process_state(*proc*) = **PS_Running**) ⇒ (*newstates*(*proc*) = **PS_Running** ∨ *newstates*(*proc*) = **PS_Ready** ∨ *newstates*(*proc*) = **PS_Waiting** ∨ *newstates*(*proc*) = **PS_Suspend** ∨ *newstates*(*proc*) = **PS_Dormant**))

then

@act01 process_state ≔ process_state *newstates*

end

end