[Arm] What is the meaning of the state running (...)?

2025-07-31 - Comments (0) - Arm

Cortex-M

running (power down)	Shown if low power mode debugging is enabled. This status is displayed when **SYStem.Option PWRDWNRecover** is set to **ON** and the target doesn't respond at all over JTAG/SWD. With **SYStem.Option PWRDWNRecover OFF** a debug port fail would be reported.
running (secured)	Chip security is enabled, which affects all cores
running (clock down)	Low power mode that turns off the clock on chip level
running (reset)	Reset on the core is asserted
running (core secured)	Debug interface secured on core level. Other cores may not be affected
running (core power down)	Core is powered down. Other cores may not be affected. State is checked by reading the CPUID register of the Cortex-M core. If reading this register returns an error or an invalid value, then this state is displayed.
running (bus error)	Cortex-M bus not accessible
running (deep sleeping)	Core is in sleep deep state indicated by the Cortex-M debug registers (DHCSR register), memory access on Cortex-M bus is blocked
running (sleeping)	Core is in sleep state indicated by the Cortex-M debug registers (DHCSR register), memory access on Cortex-M bus still allowed
running (secure zone)	Core is executing code in secure zone (TrustZone), but debugging is not allowed there
running (locked up)	Core entered an illegal state and stopped working (locked up). Main reason is, that during execution of an HARDFAULT handler another exception occurs.

Armv8/v9 Cortex-A/R

running (power down)	The core is not powered.
running (secured)	Core is secured. Access by debugger not possible. Armv8.3 DoPD is implemented.
running (sticky reset)	Core is running with sticky reset bit set.
running (OS lock)	Core is running with OS-lock bit set.
running (OS lock/catch)	Core is running with OS-lock bit set and OS catch event is enabled for this core when **SYStem.Option.BreakOS** is **OFF**.
running (no CTI)	CTI not accessible, but DAP accessible (probably cluster powered down), Armv8.3 DoPD not implemented
running (no CPU)	CPU not accessible via CoreSight (probably cluster powered down), Armv8.3 DoPD not implemented
running (no CTI DoPD)	CTI not accessible, but DAP accessible (probably cluster powered down), Armv8.3 DoPD implemented
running (no CPU DoPD)	CPU not accessible, but CTI accessible (probably cluster powered down), Armv8.3 DoPD implemented
running (disabled)	The debugger detected that CSW.DeviceEn is low. The debugger assumes the device is currently suspended. It will wait up to 10 minutes for the device to become available.
running (CTI hlt)	The core is not powered and the CTI is accessible. A halt event is pending, waiting to stop the CPU. Armv8.3 DoPD is not implemented.
running (CTI DoPD)	The core is not powered and the CTI is accessible. A halt event is pending, waiting to stop the CPU. Armv8.3 DoPD is implemented.

Armv7 Cortex-A/R

running (power down)	The core is not powered. This is detected by the core's PRSR register or can be detected on some chips that have additional control logic that provides this information. This status is also displayed when the core is temporarily not powered (detected if the **SYStem.Option.PWRDWNRecover** can be used on this device)
running (clock down)	The core is not clocked. This can be detected on some chips that have additional control logic that provides this information.
running (secured)	The core is secured. This can be detected on some chips that have additional control logic that provides this information.
running (standby WFI)	The core is in wait-for-interrupt (WFI; low power) state. This can be detected on some chips that have additional control logic that provides this information.

running (standby WFE)	The core is in wait-for-event (WFE; low power) state. This can be detected on some chips that have additional control logic that provides this information.
running (standby)	The core is in a low power state. This can be detected on some chips that have additional control logic that provides this information.
running (secure world)	The core is in the secure world and debugging in secure world is not allowed on this device. This is detected by the core's DSCR register on cores that support TrustZone (Cortex-A).
running (GPMC-WAIT0)	The used chip includes a module called GPMC (general purpose memory controller), which allows waiting for a slow memory device response by holding off bus activity. When the GPMC function is enabled, the device activates a wait input and keeps it asserted. In this state, any attempt to stop the core will be ignored. Only once the WAIT input is disabled, the debugger will become able to control the CPU again. This status is displayed when the core is out of control due to this GPMC bus freeze.
running (GPMC-WAIT1)	See GPMC-WAIT0
running (mon fail)	Monitor error when TrkMON is selected for debugging (SYStem.MemAccess TrkMON)
running (OS lock)	Debug operation is currently not possible because the operating system has temporarily locked the debug register probably in order to restore the debug register after a power recovery. This is detected by the core's OSLSR register on chips supporting this mechanism, e.g. Cortex-A7/A15/A17.