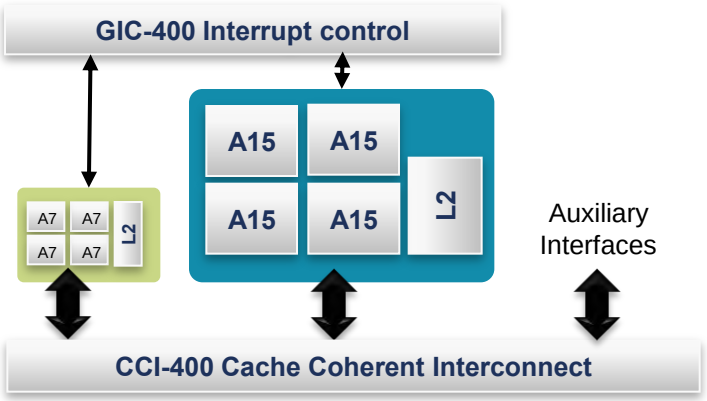
Idle Power States Nomenclature

charles.garcia-tobin@arm.com

The Architecture for the Digital World®

ARM Systems

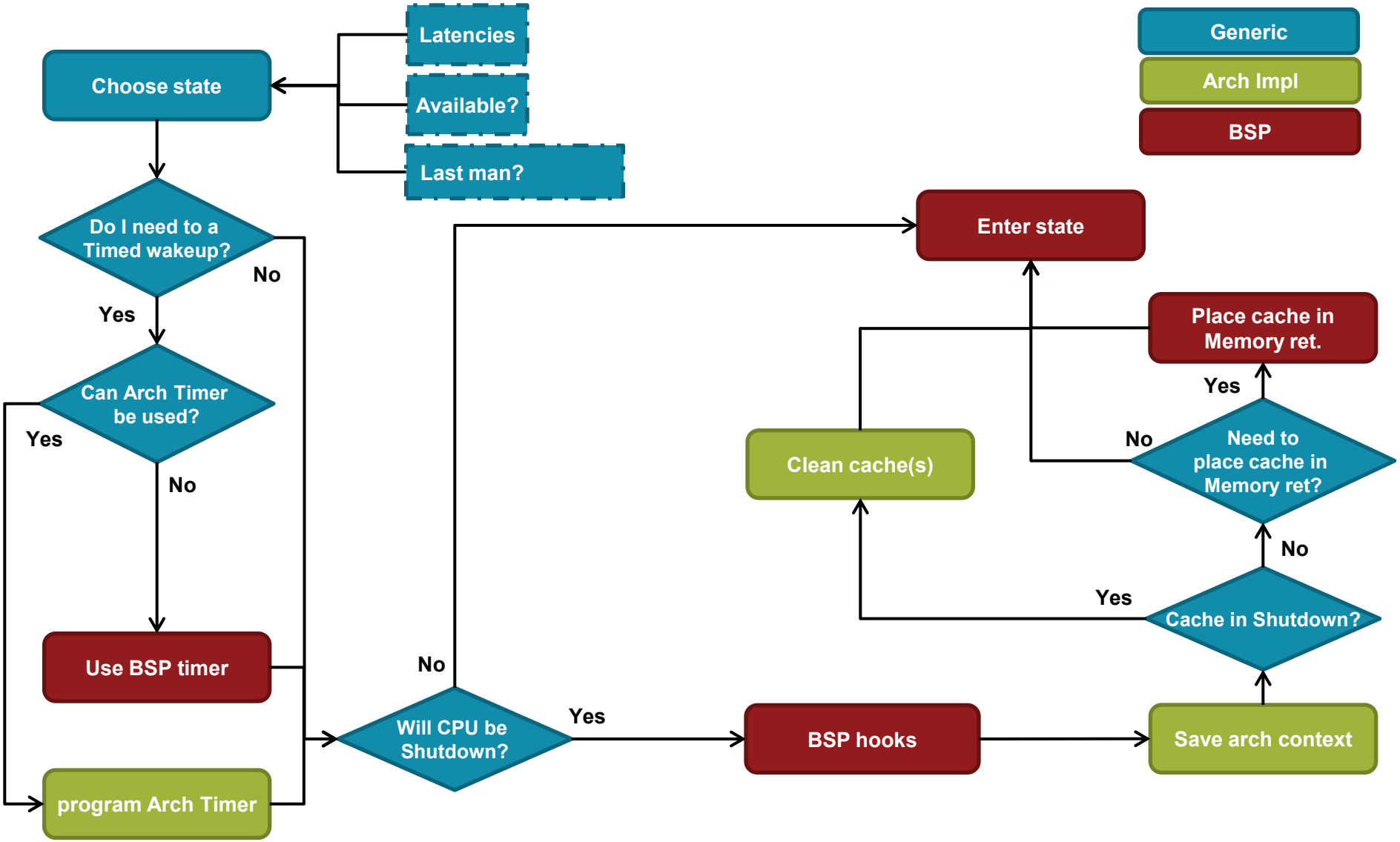
- ARM systems are increasingly complex and hierarchical
  - Low power states can require cooperation between affected cores
  - Introduction of clusters gives rise to new levels of hierarchy in power states



- SoC vendors differentiate on power
  - Power modes supported will differ from one device to another, in number and type
- However there are a lots of commonalities
  - Some states retain context and others require context saving
    - Large proportion of the context is given by the ARM architecture and its implementation
  - Some states require cache management
  - Can require wake up after a period of time
  - States require communication with an external power controller

The Architecture for the Digital World®

What do you need to do to enter a state?



The Architecture for the Digital World®
ARM®

ACPI – For better or for worse

- Linux world on ARM has adopted ACPI as its nomenclature for describing idle states
- This nomenclature makes sense in the Intel world
  - C-states have tight definitions
  - There is broad equivalence between cores
- In ARM however different platforms have different amount of states and different meanings for states
  - There is no equivalence
  - Different vendors expose different number and types of states
  - Numerically states are not equivalent eg C2 for one device is different to C2 in another

The Architecture for the Digital World®

ACPI – For better or for worse

- The only hard rule is that larger numbers mean deeper states
  - Eg C2 saves more power than C1

- However there is no particular structure for state naming

Cx???

Who is going down? It it a CPU is it a cluster? Is it all the on-line cores?

Do I need to save state?

Is a cache is affected? Which one?

How is the cache affected?

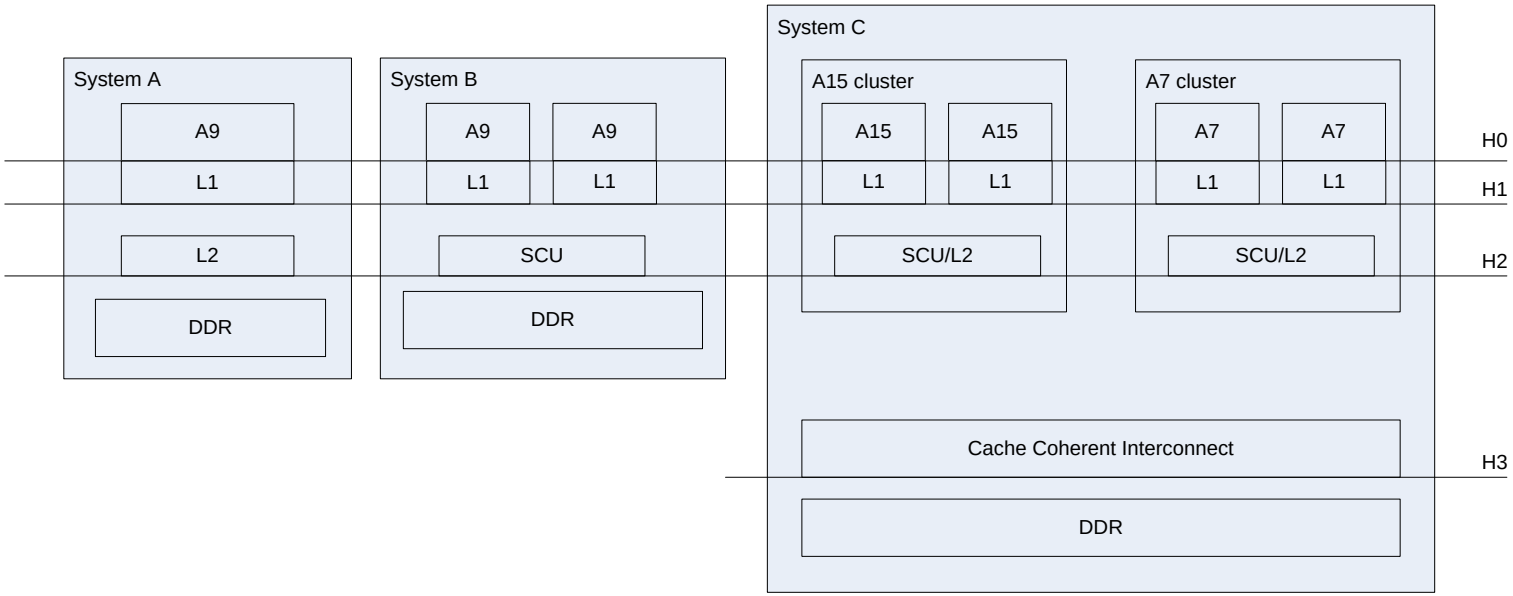
The Architecture for the Digital World®

Idle nomenclature aims and motivations

- Give common definitions that can be used across systems from different providers. Allowing comparisons to take place
- Aims to provide enough flexibility to allow differentiation
- Aim to drive further code abstraction in OS, or CPU architectural layers of OSs
- Encapsulating common OS operations
  - Which CPUs in system will be switched off -> a single core, a cluster, all cores?
  - Which caches are affected
  - When is state going to be lost? What state will be lost?
    - CPU state
    - Cache state
      - do caches need cleaning, invalidating, do they retain content
    - GIC

The Architecture for the Digital World®

Proposal - Hierarchy Levels



- A hierarchy level is bounded by a either:
  - A cache or a coherent interconnect
  - Proposal is to talk about power states affecting different hierarchy levels
    - With topology knowledge this combines affinity and cache level

The Architecture for the Digital World®

States of Execution - Running

- For a CPU (H0)
  - The CPU is executing code, higher hierarchy levels (eg caches and interconnects requires to support this CPU) are also running
- For a cache or interconnect (Hx>0)
  - The cache or interconnect at the hierarchy level is fully operational

The Architecture for the Digital World®

States of Execution - Waiting

- For a CPU (H0)
  - The CPU not executing code (STANDBYWFI). All hierarchy levels > 0 are running or waiting
  - There is no loss of state from OS point of view. OS does not have to save context
  - CPU resumes execution at the instruction after the WFI.
  - Can include clock gating and retention techniques

- For bigger (Hx>0) hierarchies bounded by a cache
  - Caches can be entered into low power states that retain memory content
  - As cache content is coherent the cache must be snoopable
    - A cache in a low power state must automatically wakeup to service snoops, transparently to the CPU
    - There could be an increase in snoop latency associated with this state

The Architecture for the Digital World®

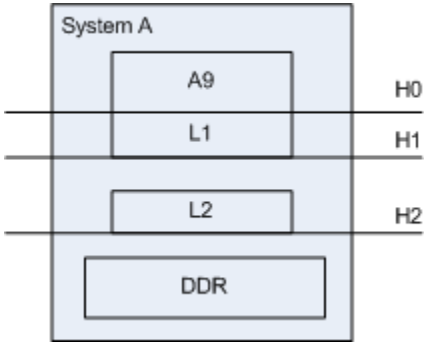
States of Execution - Shutdown

- For a CPU (H0)
  - Core is power gated
  - All CPU state is lost. OS has to save context
    - GP registers, VFP/NEON, CP15, debug state (core domain debug registers and PMUs), CPU timers
  - Resumption of execution takes place at the reset vector

- For deeper hierarchies (Hx>0)
  - The caches or interconnects contained in the hierarchy will be power gated. Any data contained will be lost
  - Caches need to be cleaned when entering the state, and invalidated when returning to a Running state

The Architecture for the Digital World®

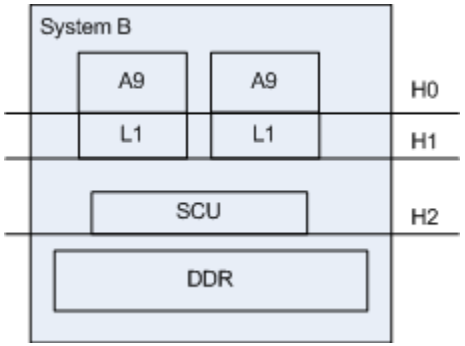
Some examples



State Type	CPU state	L1	L2	State Lost
WH0	WFI	Live	Live	None
SH1	Off	Off	Live	CPU state L1
WH2	Shutdown	Shutdown	Available	CPU state and L1 state
SH2	Off	Cleaned and Off	Cleaned and Off	CPU state L1 and L2

The Architecture for the Digital World®
ARM®

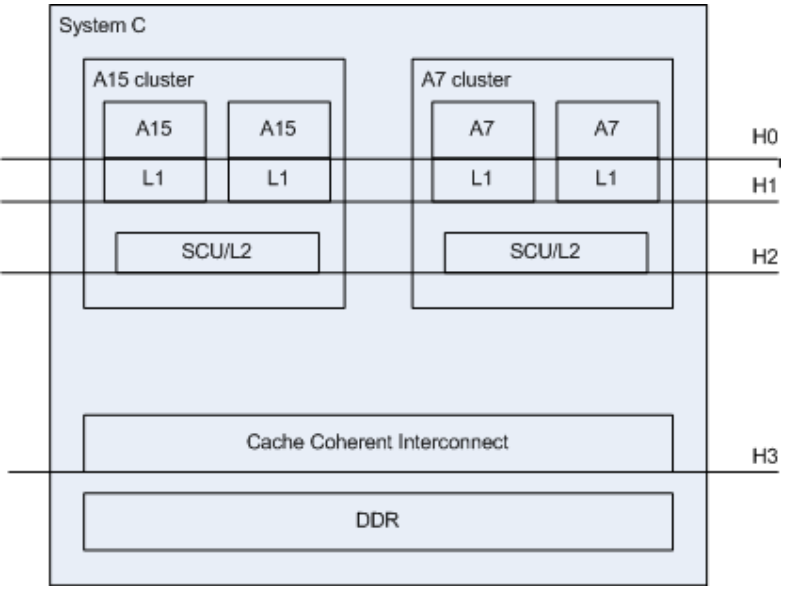
Some examples



	State Type	**State of CPUs**	**L1**	**State Lost**
CPU affinity level 0	**WH0**	WFI	Live	None
	SH1	Off	Cleaned and Off	CPU state L1 state
System affinity level 1	**SH2**	Off	Cleaned and Off	CPU state L1

The Architecture for the Digital World®
ARM®

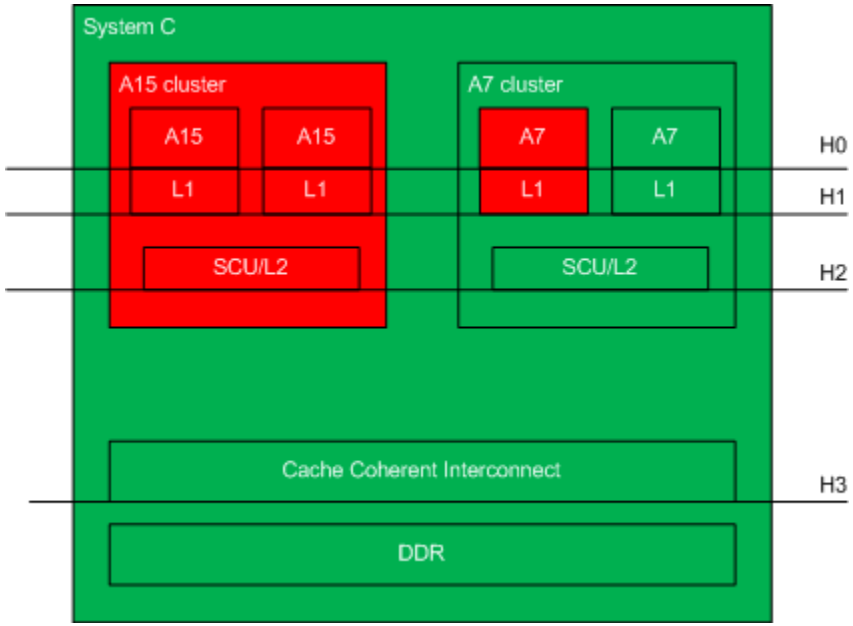
Some examples



	State Type	CPU or cluster state	L1	L2	State Lost
CPU affinity level 0	**WH0**	WFI	Live	Live	None
	SH1	Off	Off	Live	All CPU state and L1 state
Cluster affinity level 1	**WH2**	Shutdown	Shutdown	Available	CPU state L1 state
	SH2	Off	Cleaned and Off	Cleaned and Off	CPU state L1 and L2 state
System Affinity level 2	**SH3**	Off	Off	Off	CPU state L1 and L2 state

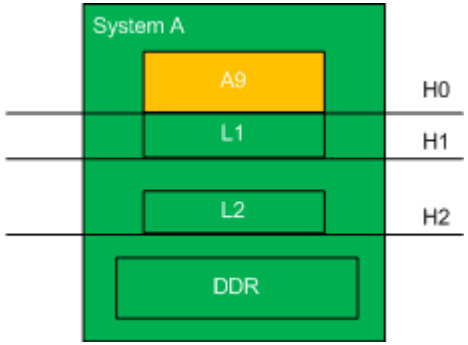
The Architecture for the Digital World®

Some examples

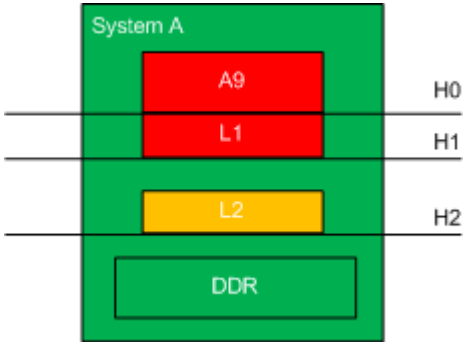


A15 cluster SH2
A7 CPU0L1 in SH1
A7 CPU1L1 in RH1

Running
Waiting
Shutdown



WH0



WH2

The Architecture for the Digital World®
ARM®

State ID

- Different HW platforms may support several states of each type
- It is proposed that states can have individual IDs appended to the name of the state eg:

[**R**/**W**/**S**][Hierarchy level]_[StateID]

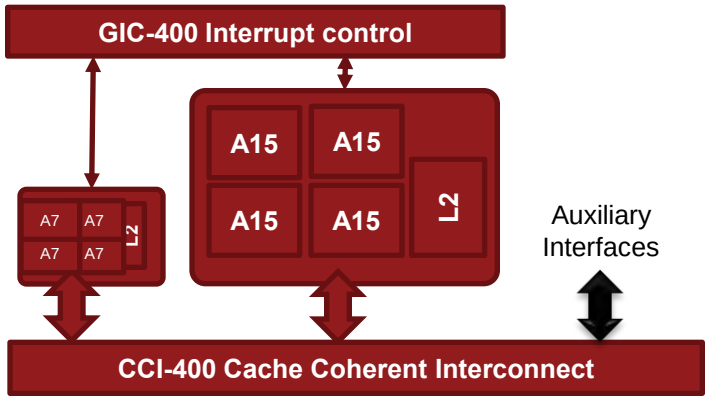
The Architecture for the Digital World®

Additional properties

- An architected framework for idle power management needs to track a number of additional properties per state
  - GIC state
  - Latencies
  - Wake up timer
  - Availability
  - Other

The Architecture for the Digital World®

Additional properties - GIC



- In some shutdown states at higher levels, it is possible to loose GIC state
- Needs to be saved
- A flag needs to be associated with the appropriate system shutdown states

The Architecture for the Digital World®
ARM®

Additional properties - Latencies

- **Entry/Exit:** OSPM needs to know the aggregate time to:
  - Enter the state
  - Move the hierarchy level back into execution

- **Memory Latencies:**
  - When a cache is in waiting, there will be a snoop latency associated with that state:
    - The OSPM should not use a cache waiting state if:
      - there are other bus masters which are active that can snoop into the cache AND their memory quality of service requirements cannot be satisfied due to the snoop latency

The Architecture for the Digital World®

Additional properties – Wakeup Timer

- ARMv7 provides architectural timers, generic timer, that can be used to wake up cores from waiting states
- Generic Timer cannot be used in all Shutdown states
  - Software standardisation could work round this problem
- Per state we need to represent a flag to indicate if external timer wake up is required
  - If not the OSPM programs the architectural timer
  - Otherwise it calls out to the BSP to program a timer

The Architecture for the Digital World®

Additional properties - Availability

- Availability of a power state is not just determined by latency
- Current mode of other components in the system can determine availability of states:
  - E.g. GPU is running, or use of some clocks
  - Device power management within the OS can be used to gate states
- big.LITTLE Migration models also introduce the concept of per CPU idle states

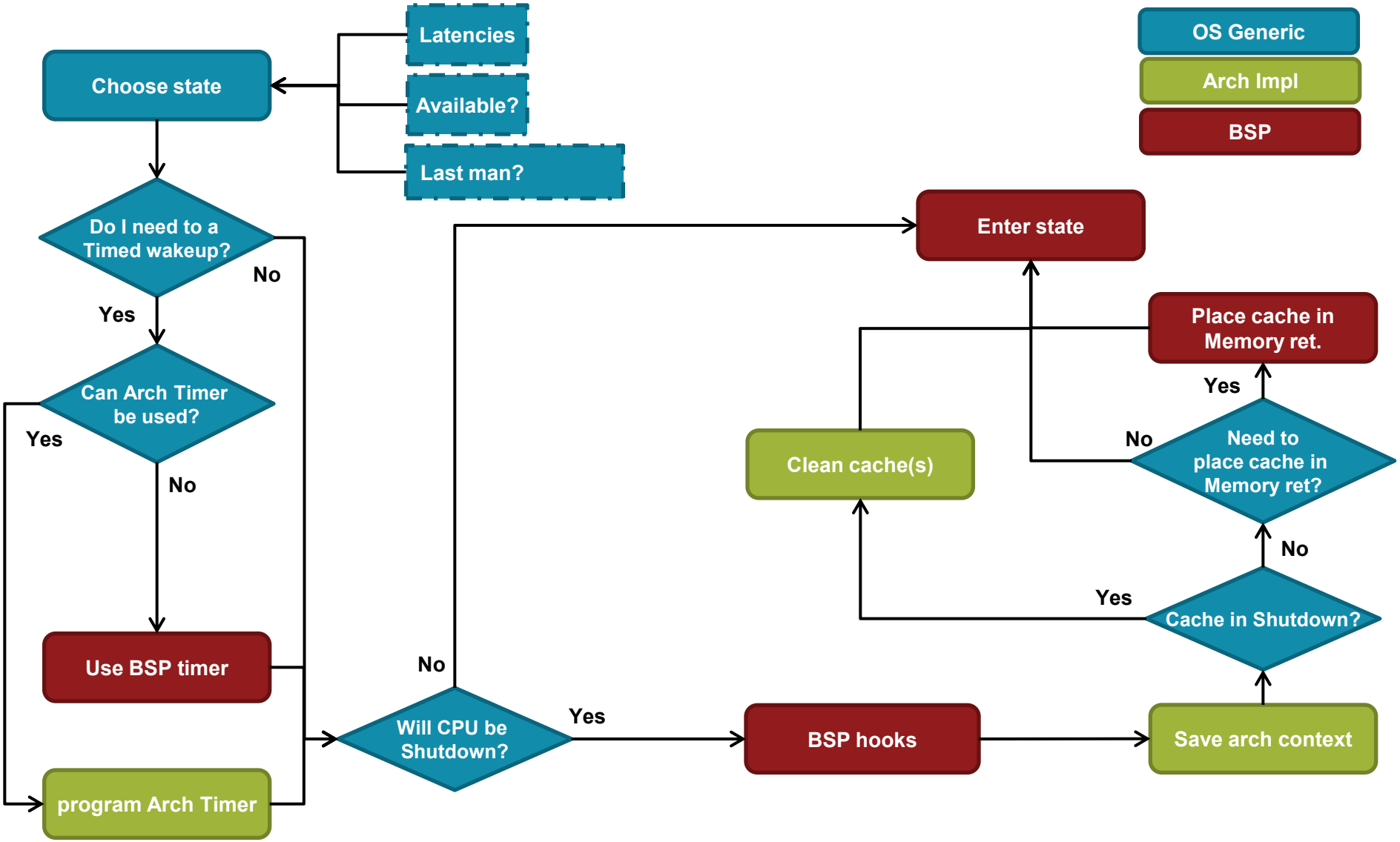
The Architecture for the Digital World®

Additional properties – Other

- Last Man
  - In some systems (mainly owing to affinitised trustedOS) only one specific CPU can take the cluster/system down
- Target residency
- Power consumed by state

The Architecture for the Digital World®

Putting it all together



The Architecture for the Digital World®
ARM®

Questions?

The Architecture for the Digital World®
ARM®