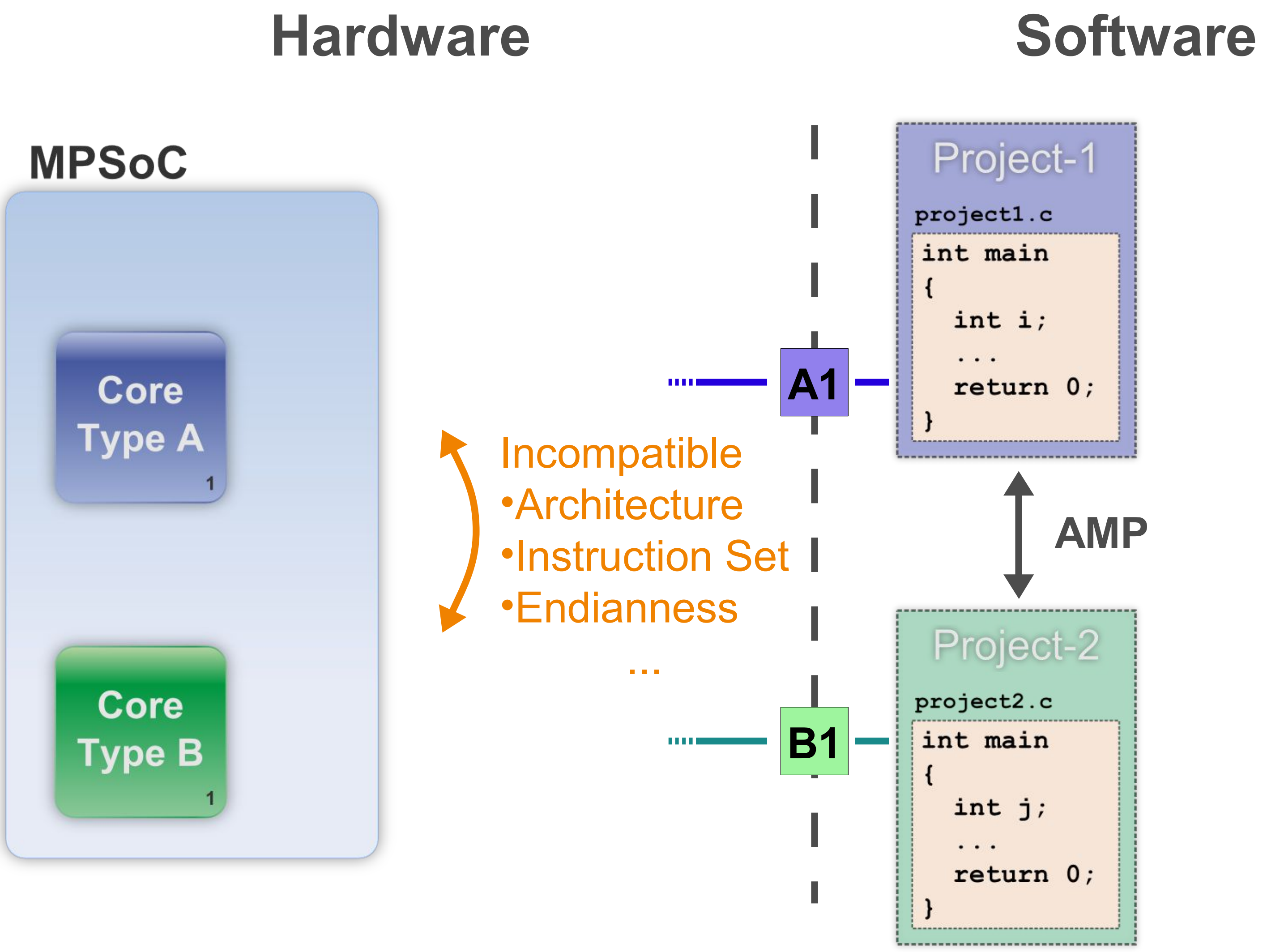
State-of-the-art Multicore Debugging and Tracing concepts

by
Alexander Merkle, Lauterbach GmbH

Agenda

- ▶ **AMP or SMP introduction**
- **Debug**
  - Hardware aspects for off-chip debug
  - Operating system points of view
- **Trace**
  - Hardware aspects for off-chip trace
  - Trace timestamping

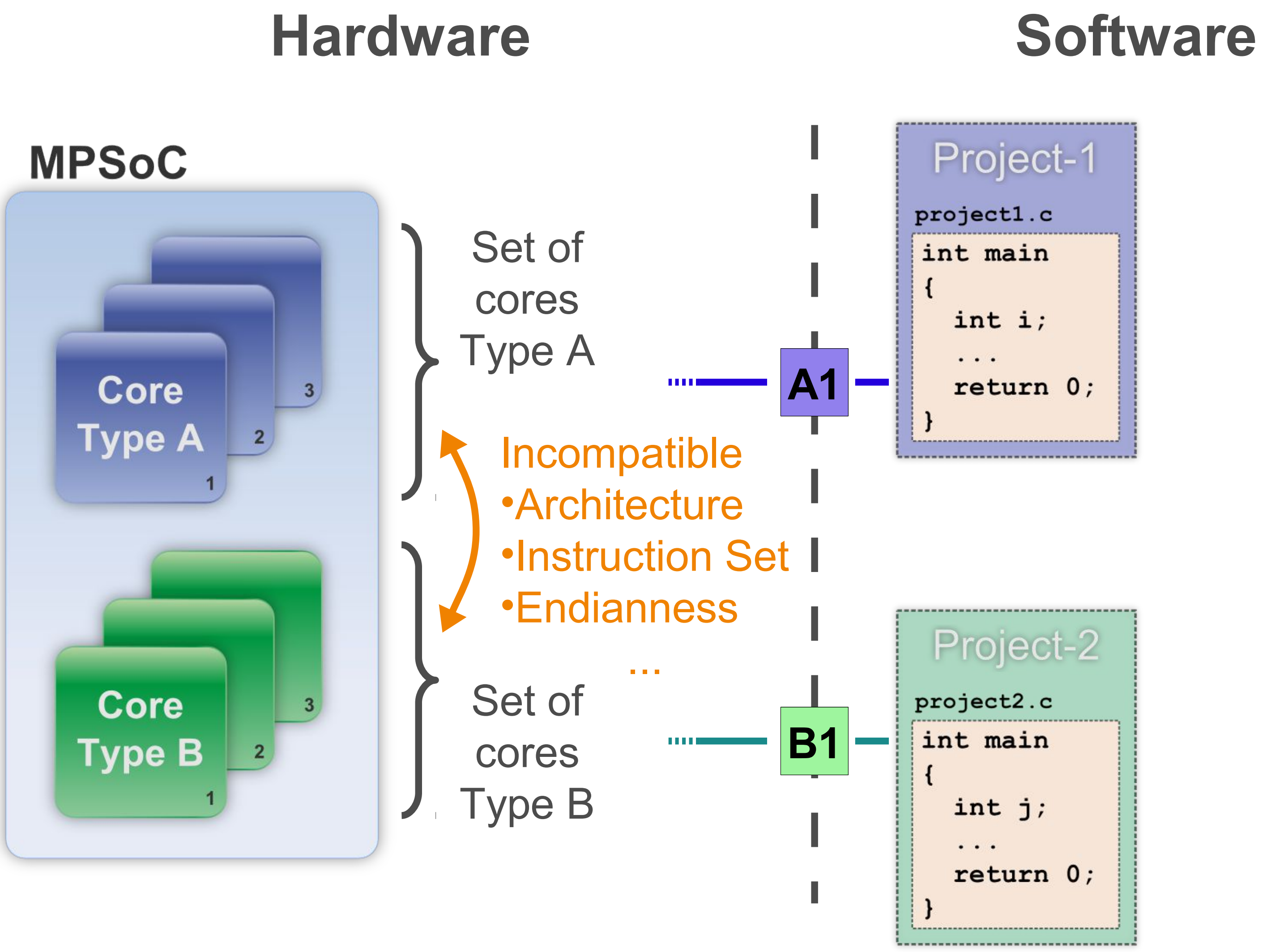
AMP – Asymmetric MultiProcessing



```
int main
{
  int i;
  ...
  return 0;
}
```

```
int main
{
  int j;
  ...
  return 0;
}
```

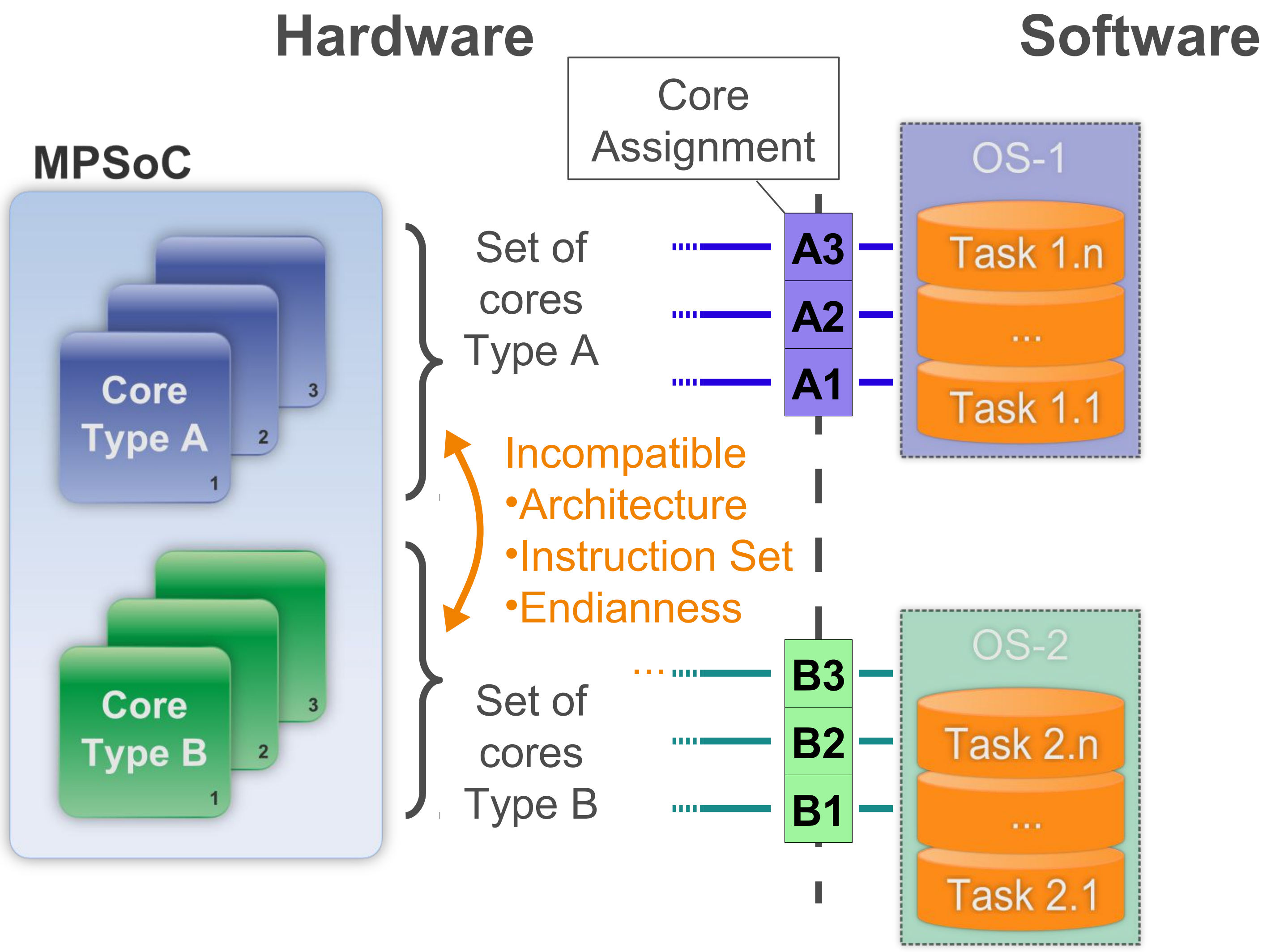
AMP – Asymmetric MultiProcessing



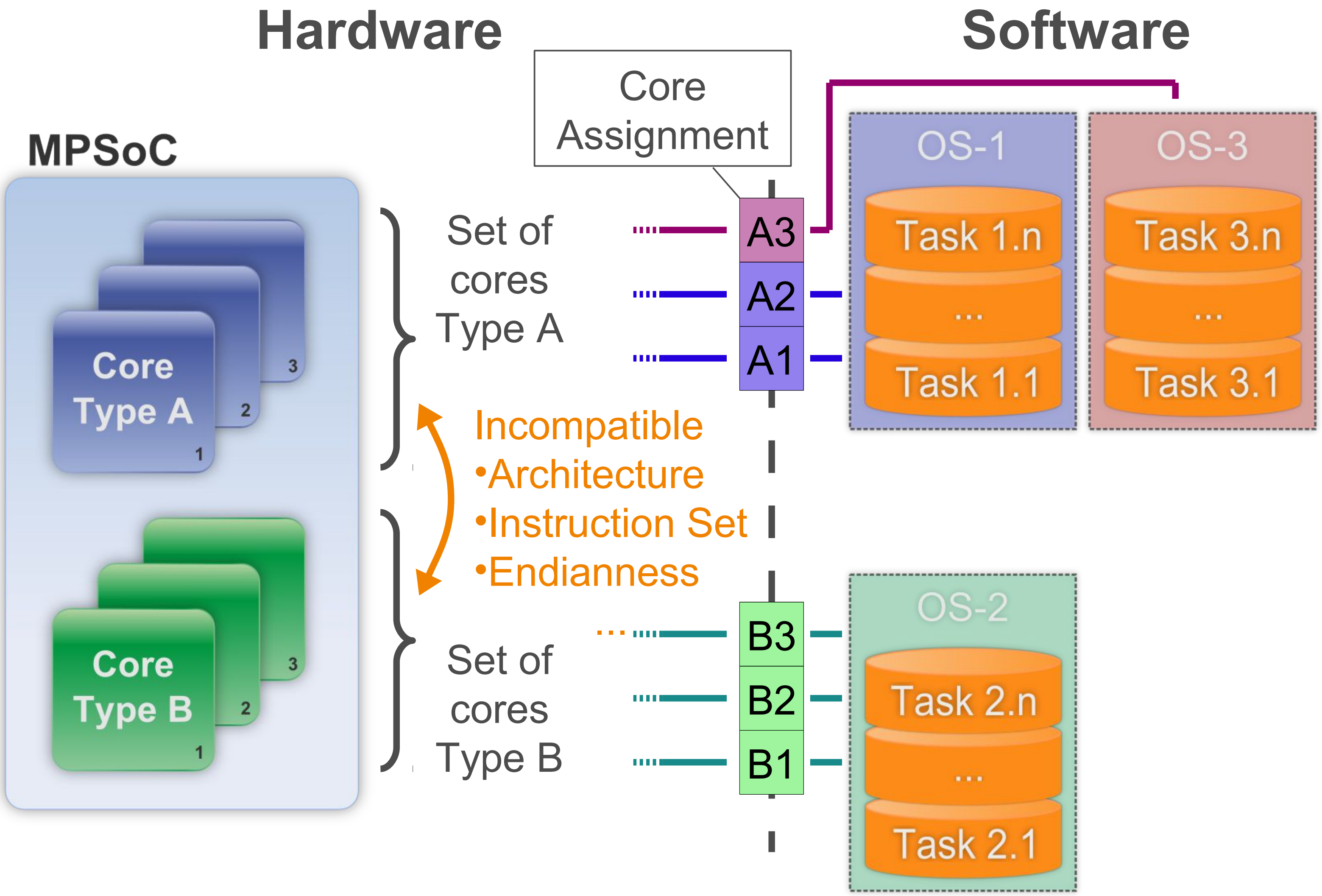
```
int main
{
  int i;
  ...
  return 0;
}
```

```
int main
{
  int j;
  ...
  return 0;
}
```

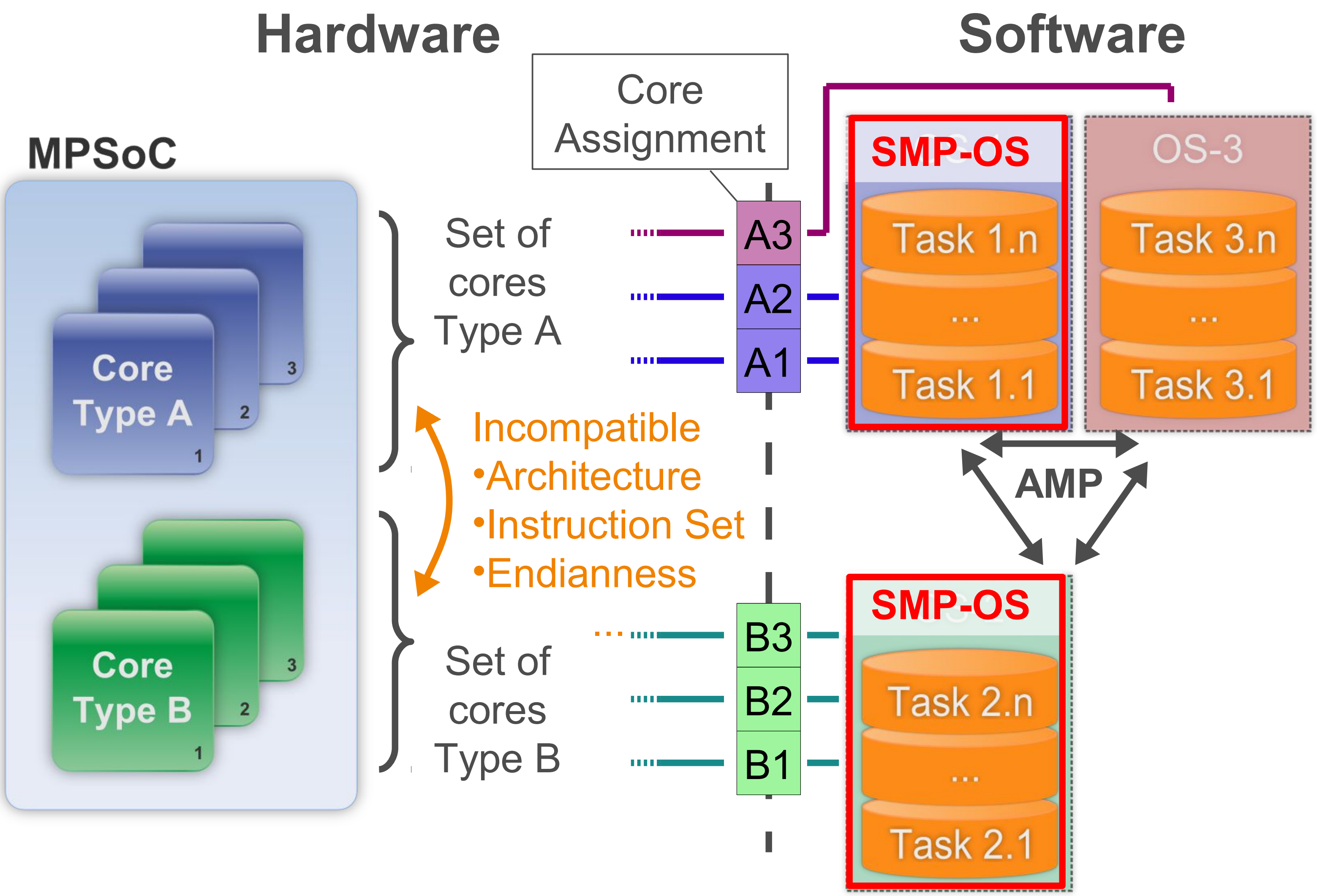
SMP – Symmetric MultiProcessing



SMP – Symmetric MultiProcessing



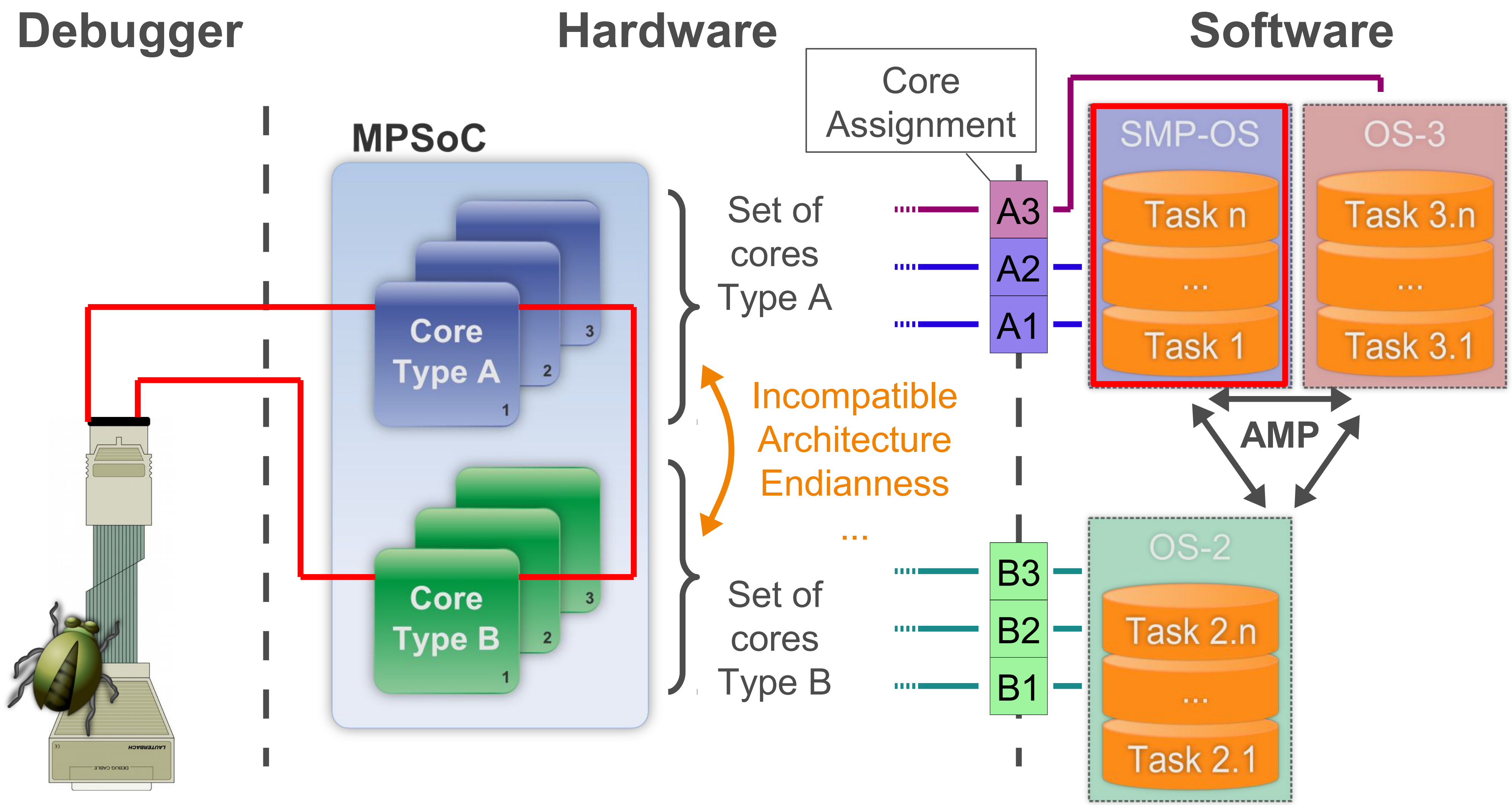
Mixed SMP and AMP configuration



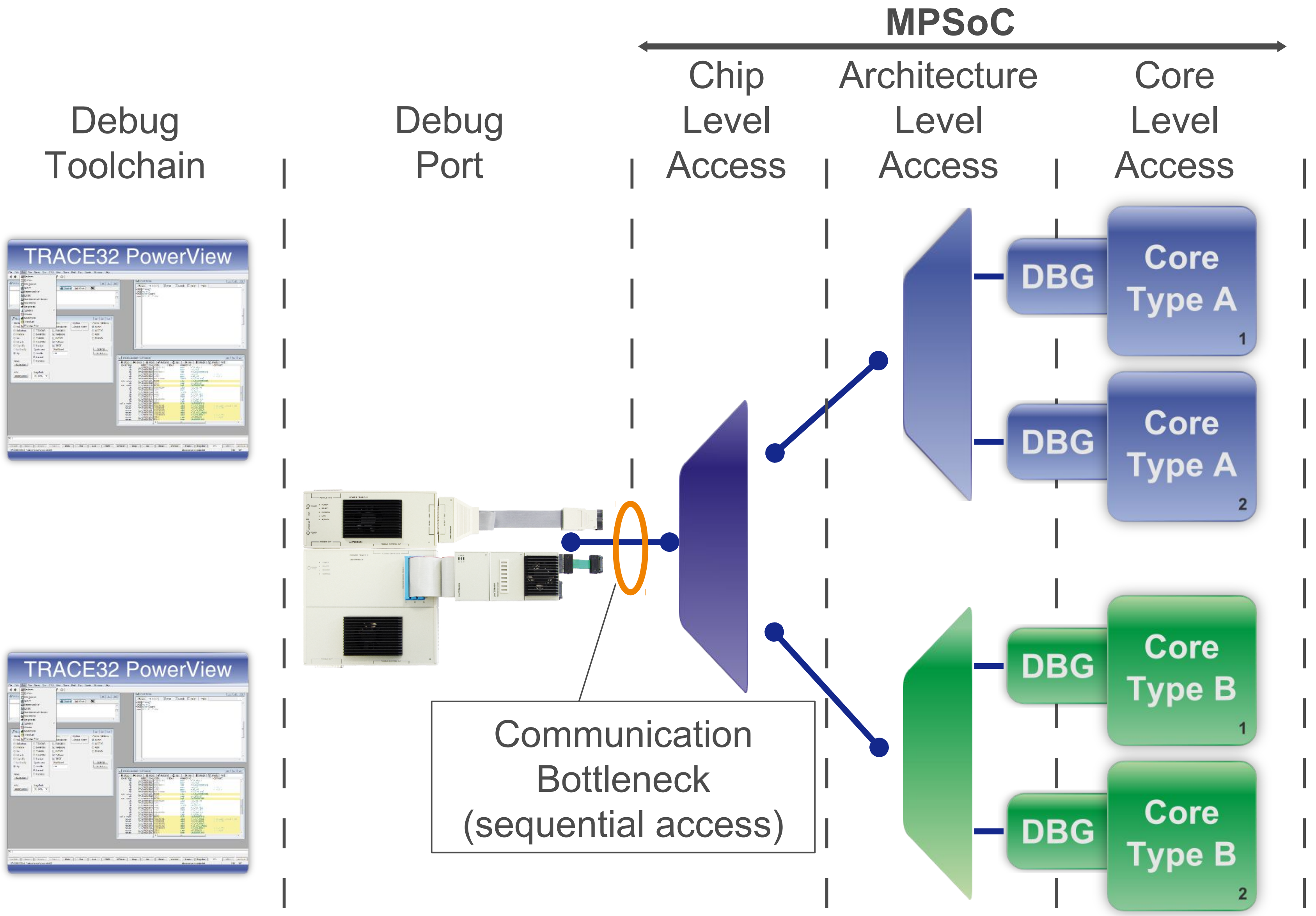
Agenda

- AMP or SMP introduction
- ▶ **Debug**
  - Hardware aspects for off-chip debug
  - Operating system points of view
- **Trace**
  - Hardware aspects for off-chip trace
  - Trace timestamping

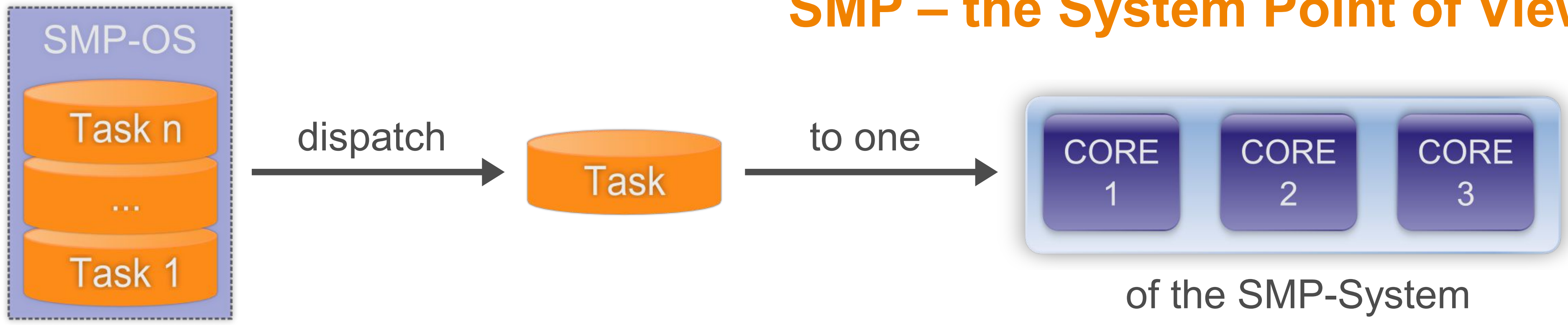
AMP or SMP – the offchip debuggers Point of View



AMP or SMP – the offchip debuggers Point of View

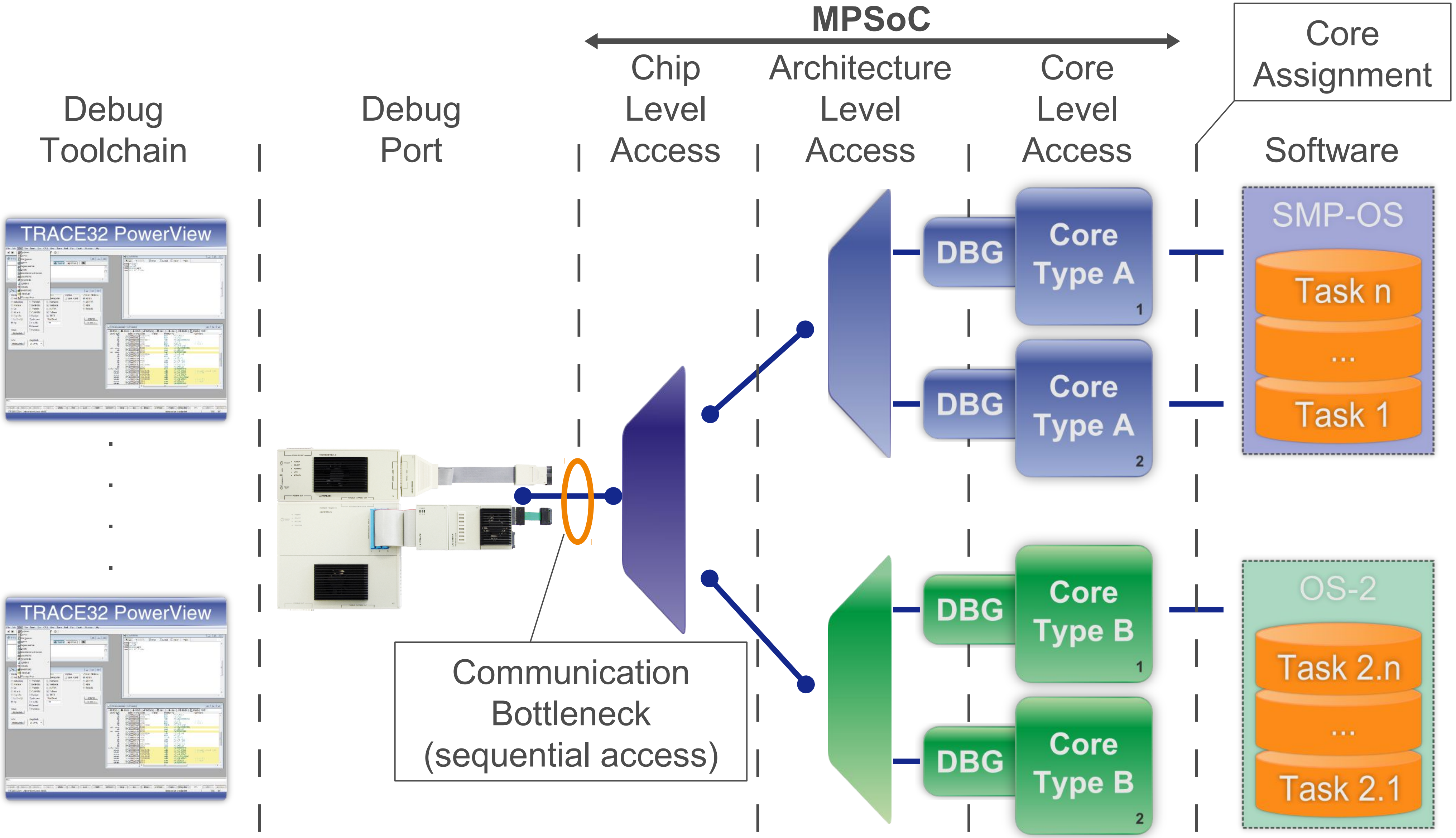


SMP – the System Point of View



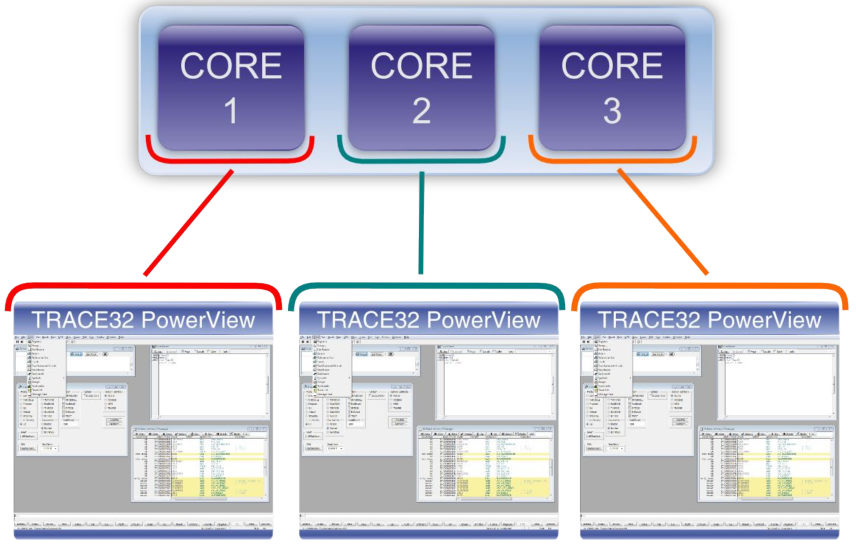
- The SMP-Operating-System (OS) dispatches TASKs to COREs.
- As all cores are equal the core to which the task is dispatched is dynamic.
- In case of SMP we need to look to the SMP-SYSTEM in total
  - Debug features must be synchronous to the whole SMP-SYSTEM
    → debugging must be synchronous on all cores
    → onchip hardware assistance for synchronous Go/Break required
  - The external debug tool needs to be **aware** of the OS and the OS core assignment.

AMP or SMP – the offchip debuggers Point of View



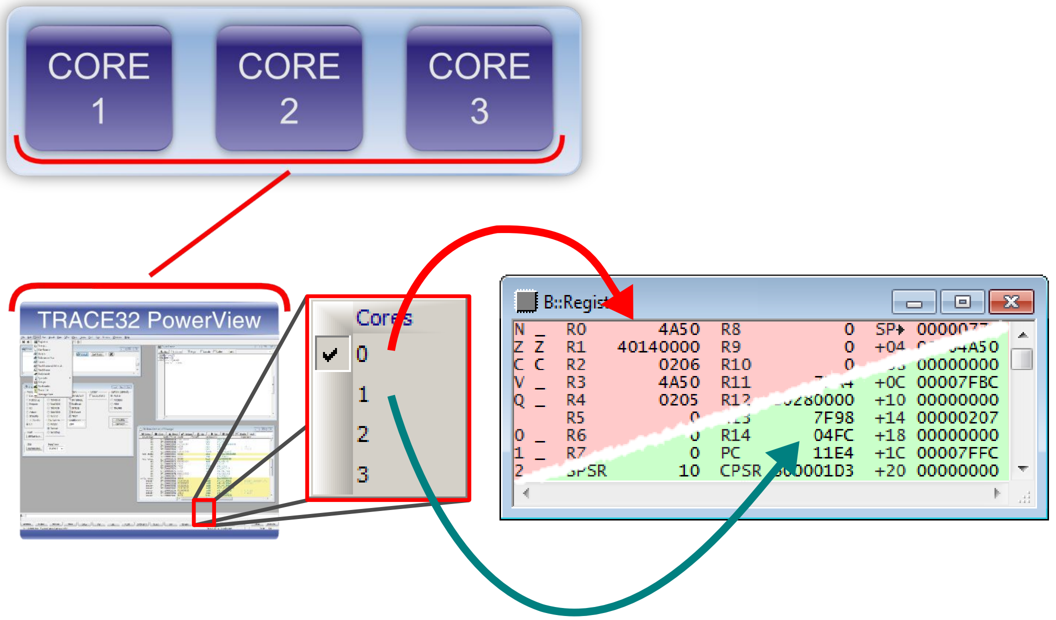
AMP / SMP debug concept

AMP



Multiple TRACE32 PowerView instances

SMP



```
N _  R0         4A50  R8            0  SP> 00000[illegible]
Z Z  R1     40140000  R9            0  +04 [illegible]4A50
C C  R2         0206  R10           0  [illegible] 00000000
V _  R3         4A50  R11   [illegible]  +0C 00007FBC
Q _  R4         0205  R12 [illegible]0280000  +10 00000000
     R5            0  [illegible]    7F98  +14 00000207
0 _  R6            0  R14        04FC  +18 00000000
1 _  R7            0  PC         11E4  +1C 00007FFC
2    [illegible]SPSR  10  CPSR [illegible]00001D3  +20 00000000
```

Single TRACE32 PowerView instance

Agenda

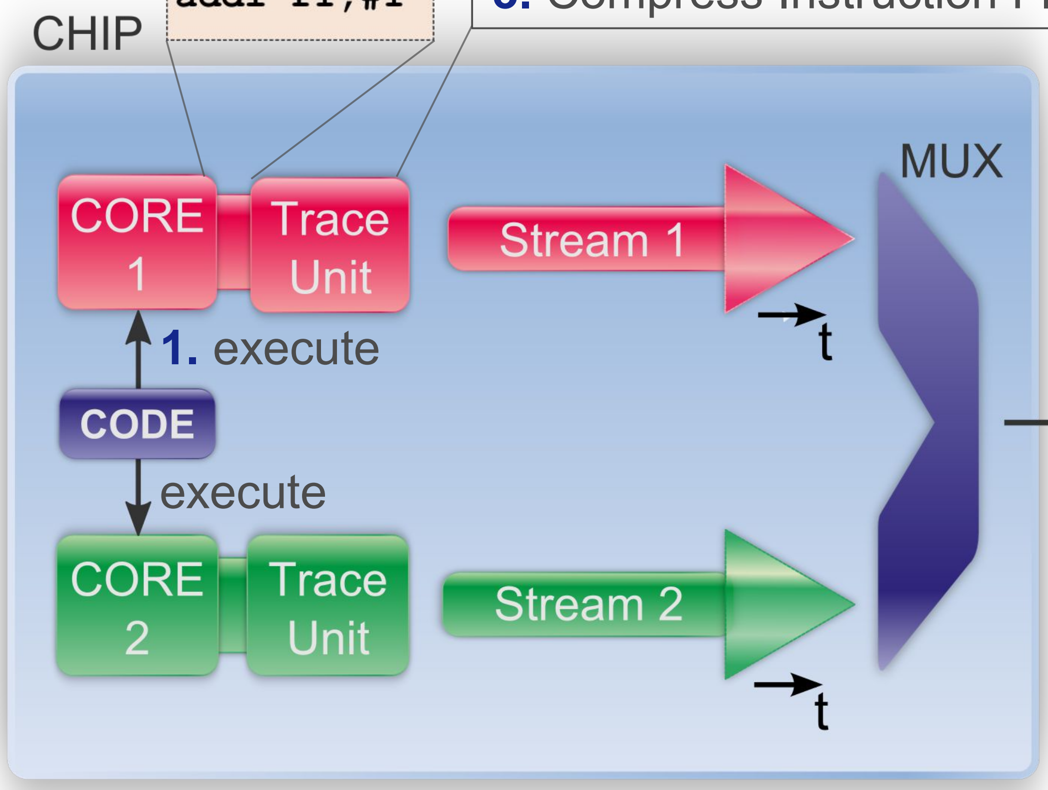
- AMP or SMP introduction
- Debug
  - Hardware aspects for off-chip debug
  - Operating system points of view
- ▶ Trace
  - Hardware aspects for off-chip trace
  - Trace timestamping

Offchip Trace - Introduction

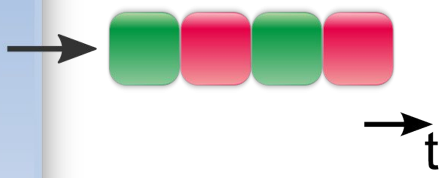
2. Instruction-Flow == executed code

```
mov   r1,r2
j     $+4
nop
addi  r1,#1
```

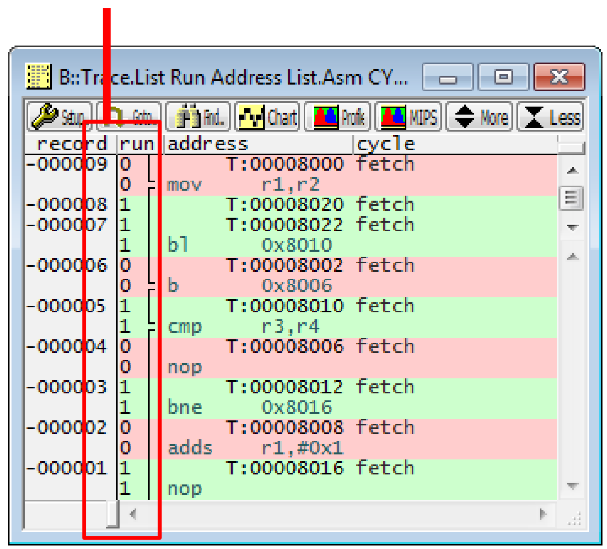
3. Compress Instruction Flow to Trace-Stream



4. Interleaved Stream 1&2



Logical core number



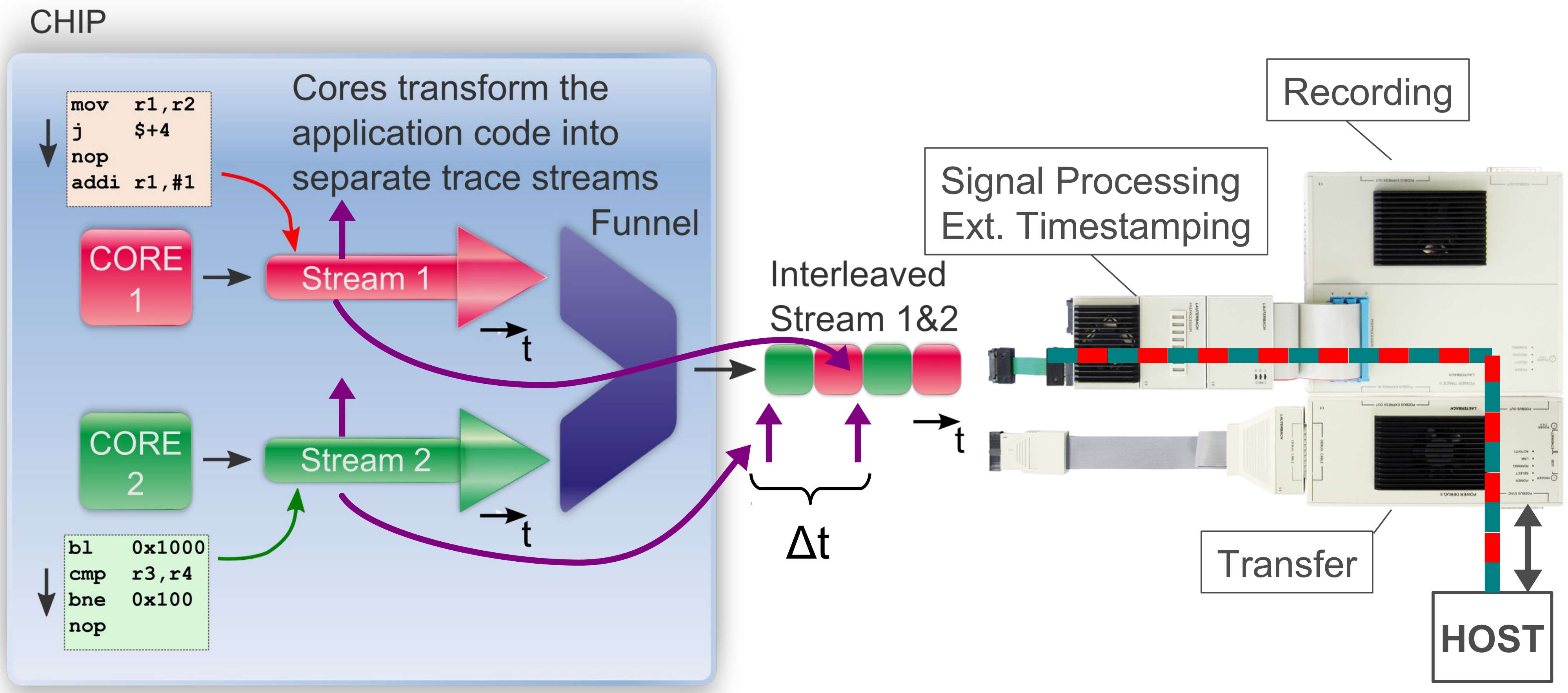
```
 record |run |address              |cycle
-000009  0              T:00008000 fetch
         0    mov         r1,r2
-000008  1              T:00008020 fetch
-000007  1              T:00008022 fetch
         1    bl          0x8010
-000006  0              T:00008002 fetch
         0    b           0x8006
-000005  1              T:00008010 fetch
         1    cmp         r3,r4
-000004  0              T:00008006 fetch
         0    nop
-000003  1              T:00008012 fetch
         1    bne         0x8016
-000002  0              T:00008008 fetch
         0    adds        r1,#0x1
-000001  1              T:00008016 fetch
         1    nop
```

Goal: reconstruct the Instruction-Flow of both cores

Trace - Timestamping

- Challenge:

  Trace data is interleaved among all cores of the chip.
  => Timing information is lost.

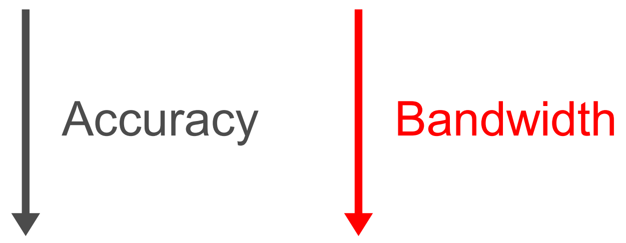


Trace - Timestamping

- Challenge:

  Trace data is interleaved among all cores of the chip.
  => Timing information is lost.

- Goal:
  - Reconstruct original „internal“ concurrent trace streams
  - Correlate the concurrent trace streams
  - Using either or a mixture of
    - Assembly level runtime interpolation
    - Chip global timestamps
    - Cycle accurate traces



- Problem:
  - Bandwidth

Trace - Outlook

- Parallel to Serial
  - Low lane count with high bandwidth.
  - Not only in High-Performance but also in Mid-Range (Realtime) market.
  - Reuse of standard peripherals like USB, PCI-Express, SATA
  - Challenges:
    - trace port is no longer optimized according to it's use-case
- System Traces
  - Instrusive but selective trace of data (software based)
  - Higher-level evaluation, protocol dependent

Conclusion

- **Multiprocessor systems use symmetric & asymmetric configurations**
- **Debugging challenges**
  - Target operating system
  - Chip/core level synchronization (Go/Break)
  - Debug port bottleneck
- **Trace challenges**
  - Trace stream correlation/timestamping
  - Trace port bandwidth

Thank you for your Attention

Questions?