



nRF51 Series introduction

- A brief look at the underlying hardware

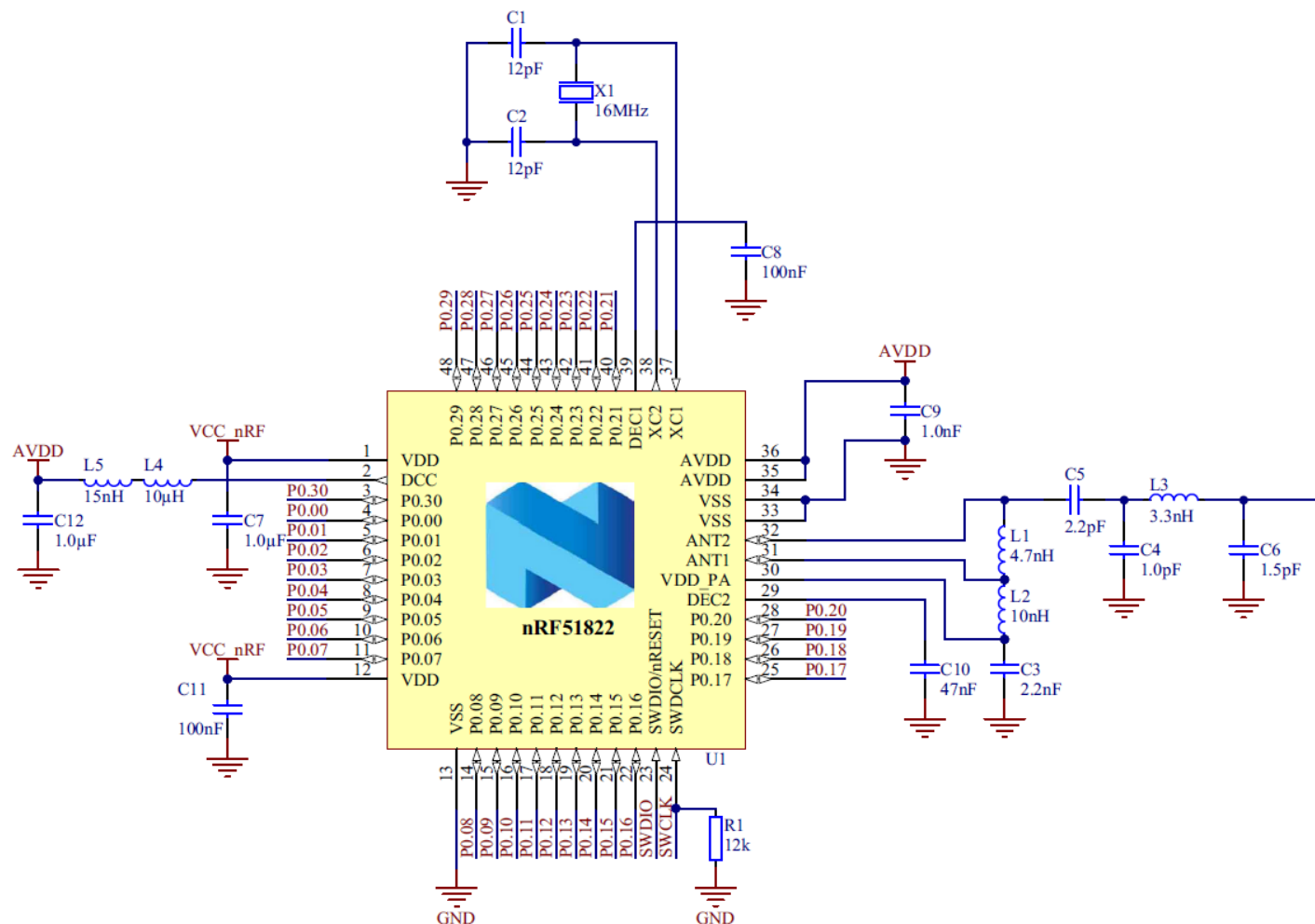


Figure 9 nRF51822-QFAA with DC/DC converter - Schematic

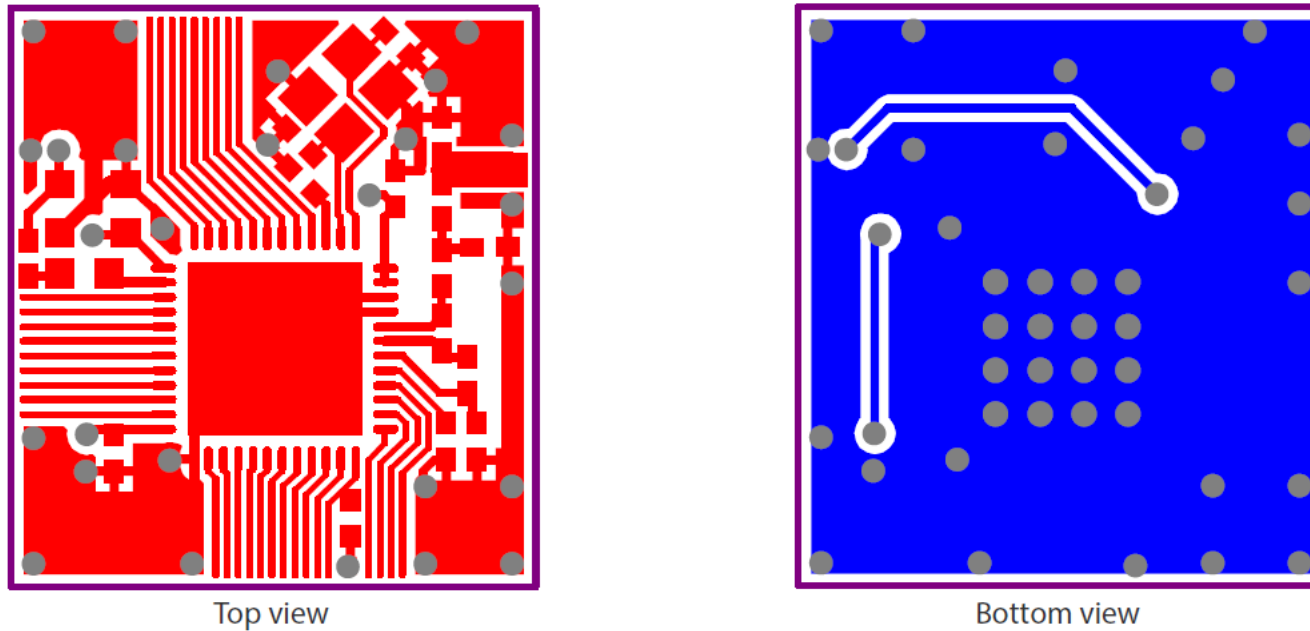


Figure 10 nRF51822-QFAA with DC/DC regulator - Layout

Goal with nRF51 Series

Flexibility

Consistency

Low Power

Goal with nRF51 Series

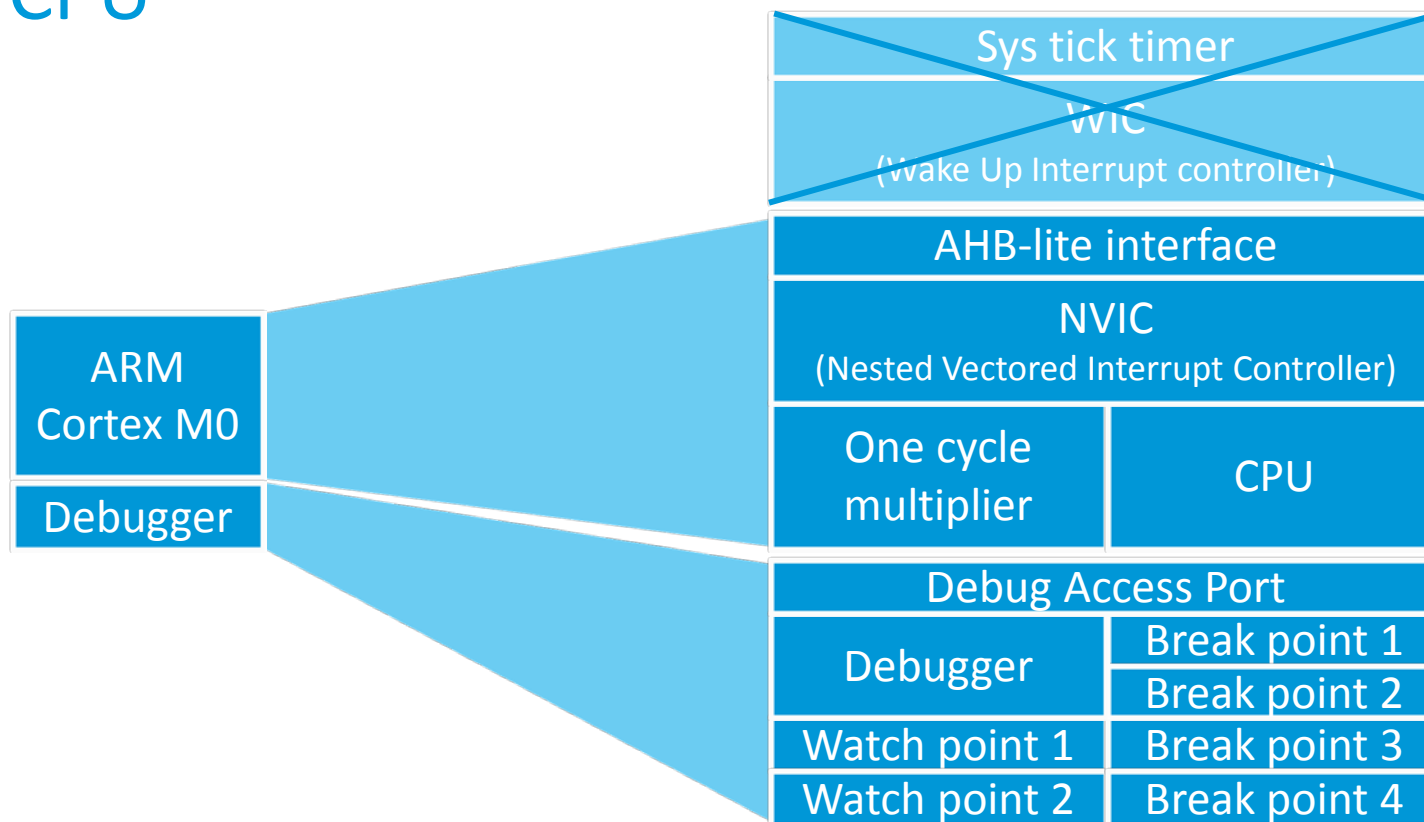
Flexibility

Consistency

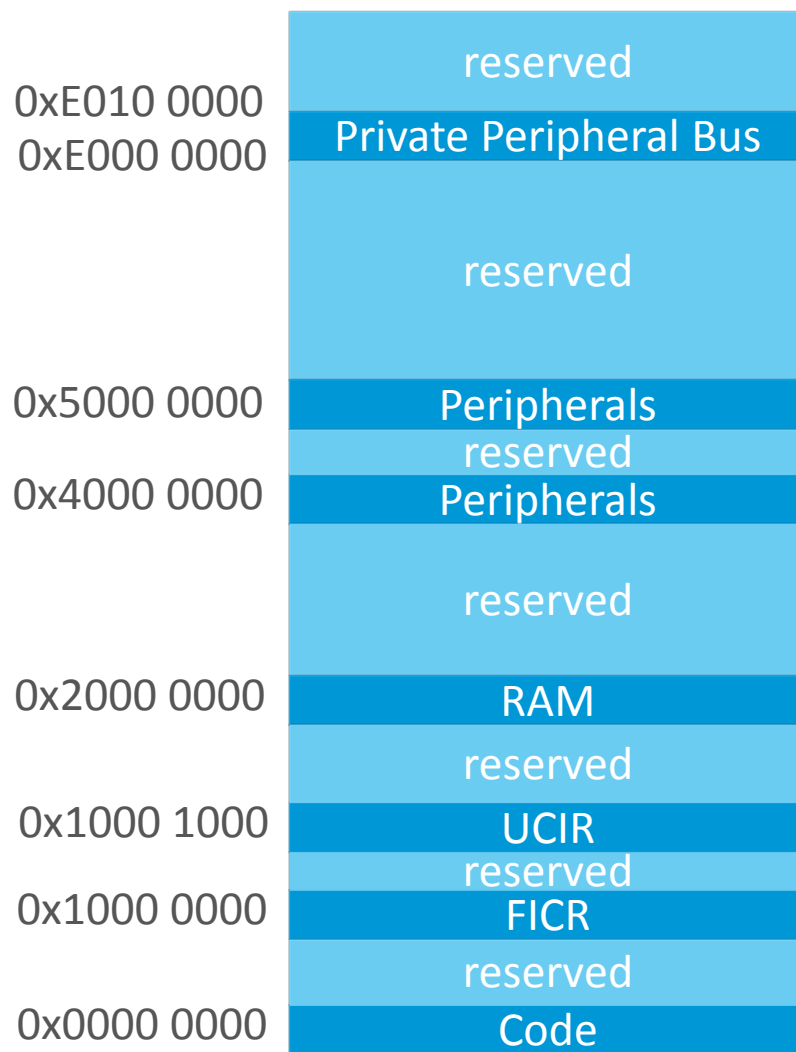
Low Power

Core functions

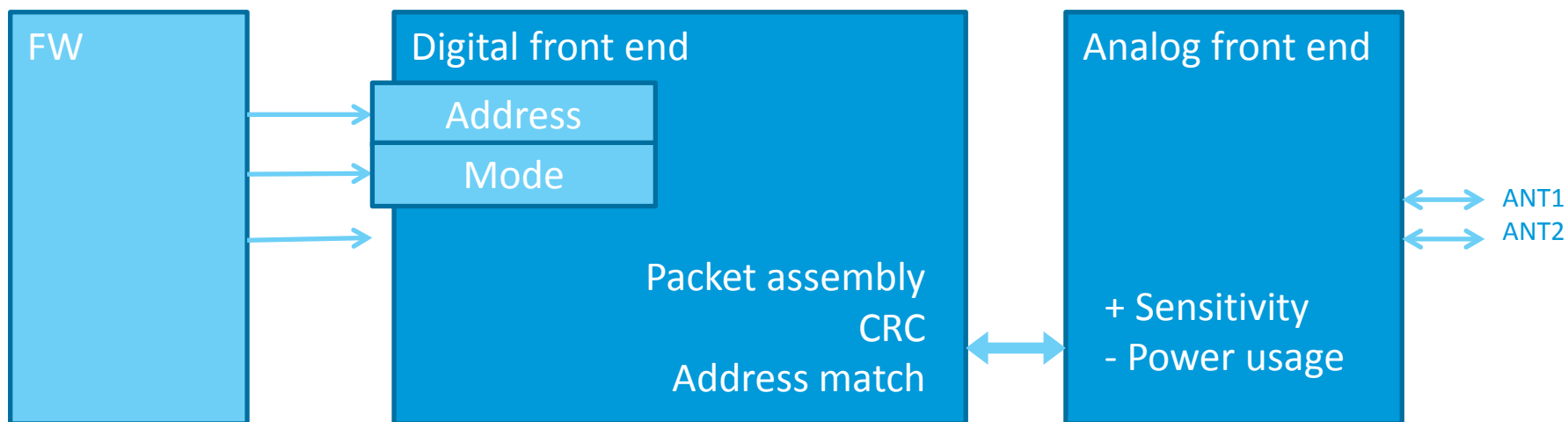
The CPU



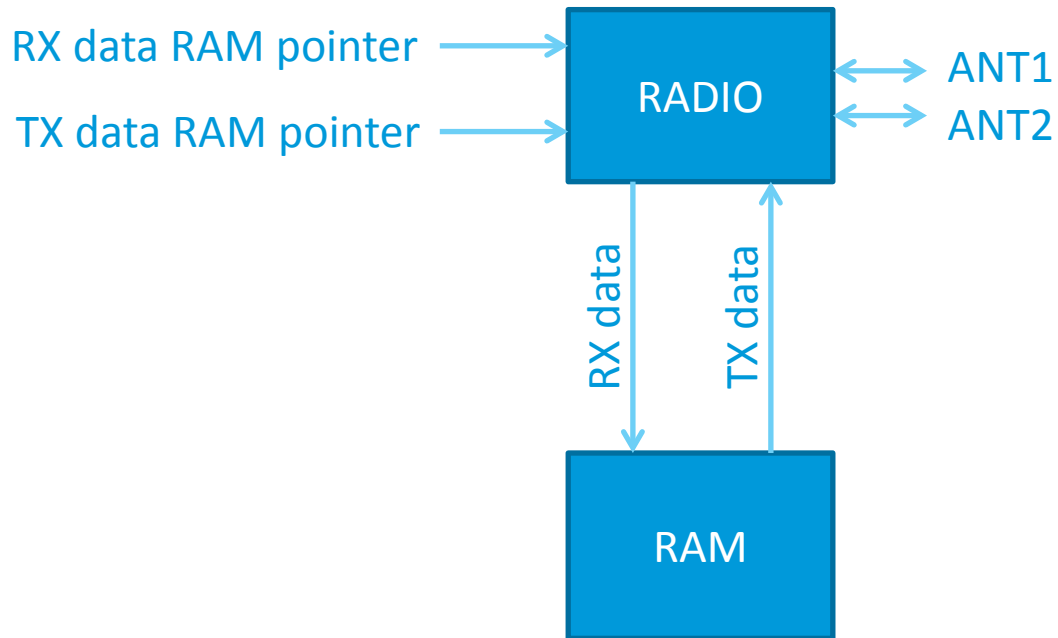
Address map



Multi Protocol Radio



Easy DMA



Goal with nRF51 Series

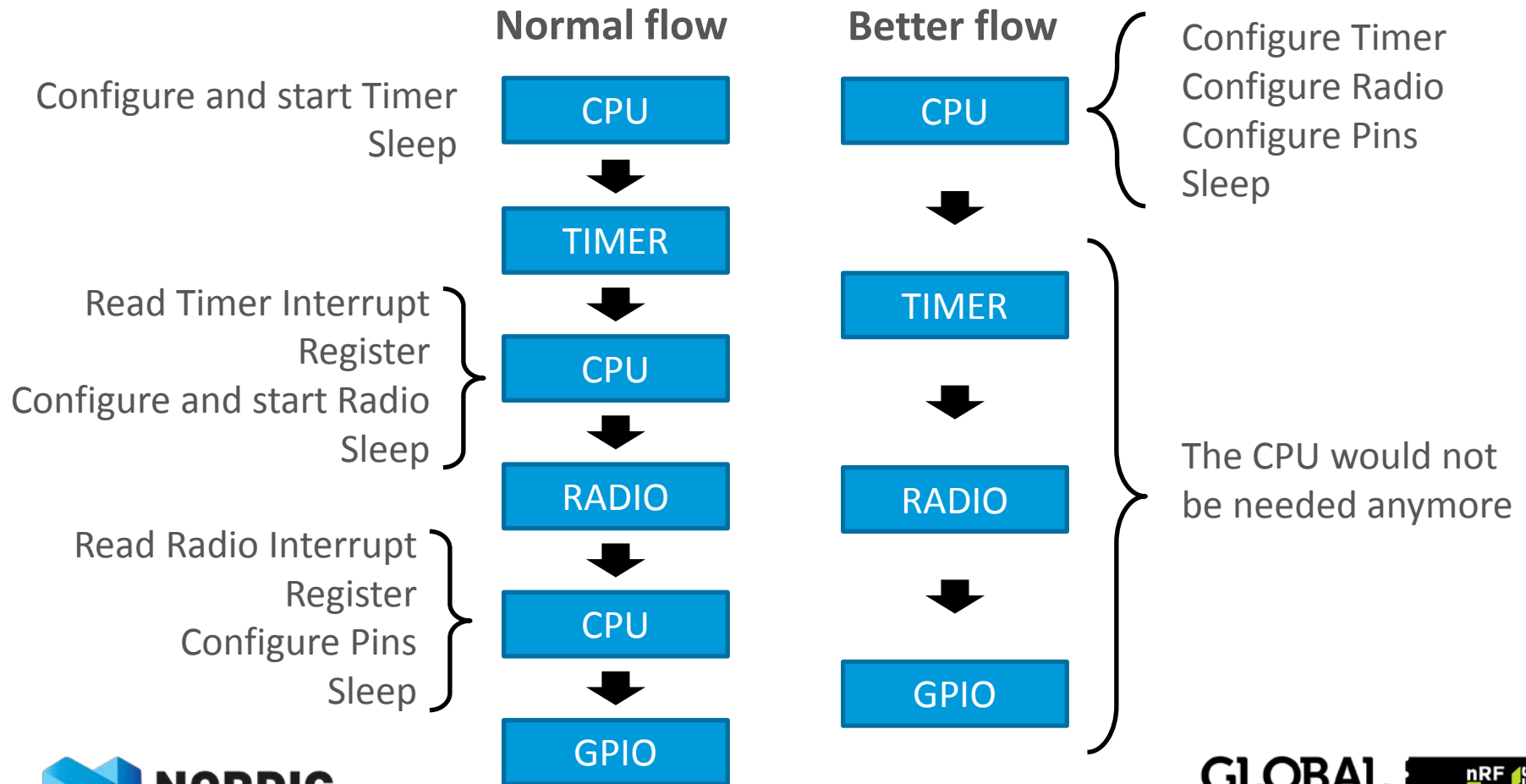
Flexibility

Consistency

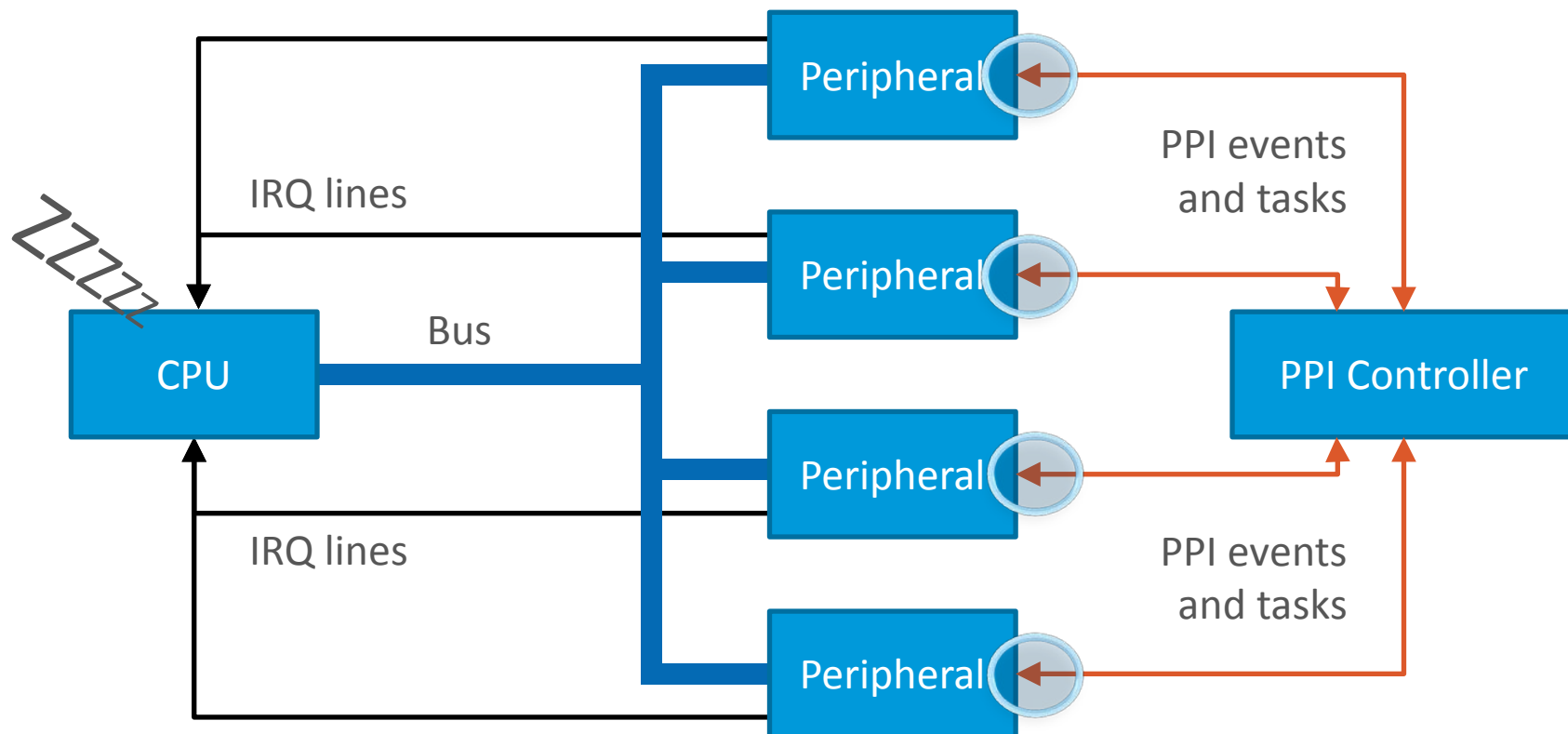
Low Power

Signalling and Connectivity

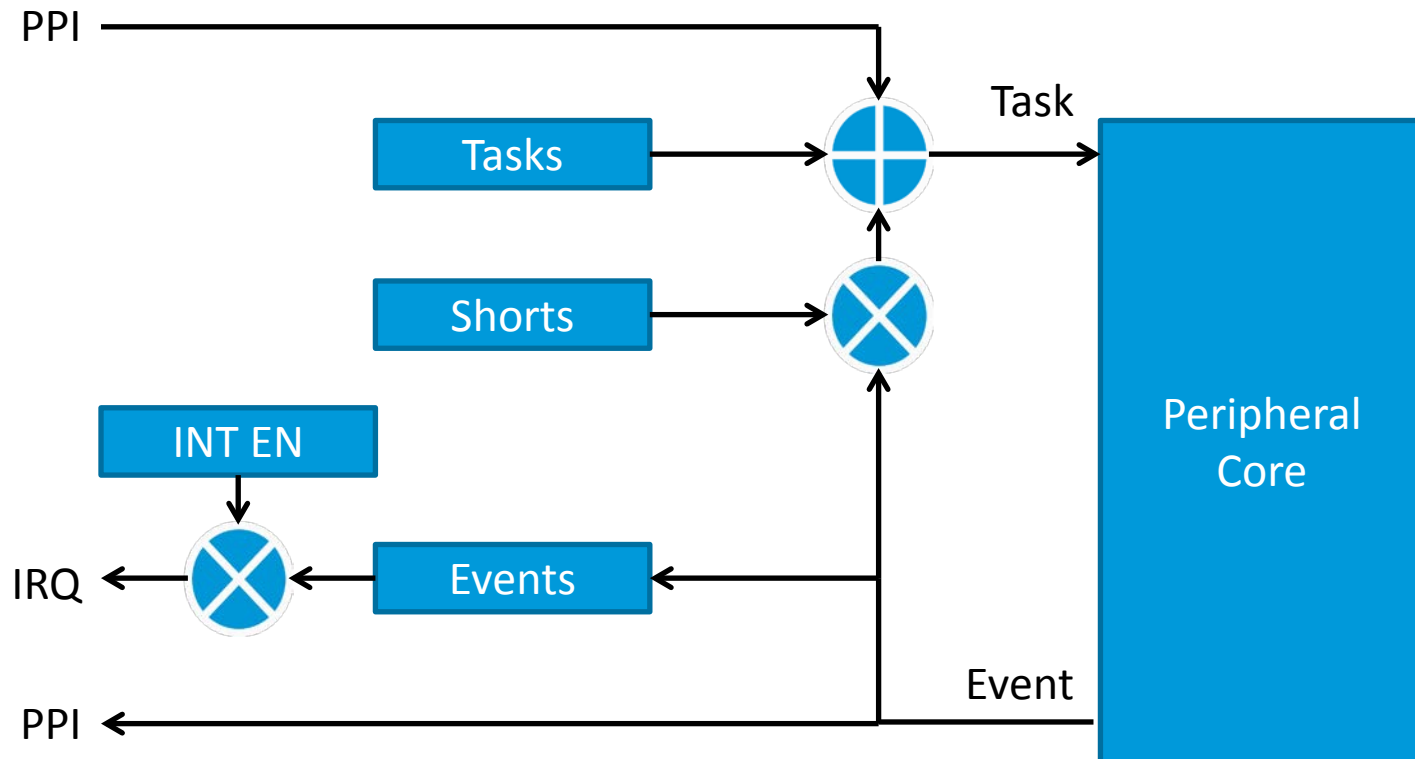
Signalling flow between peripherals



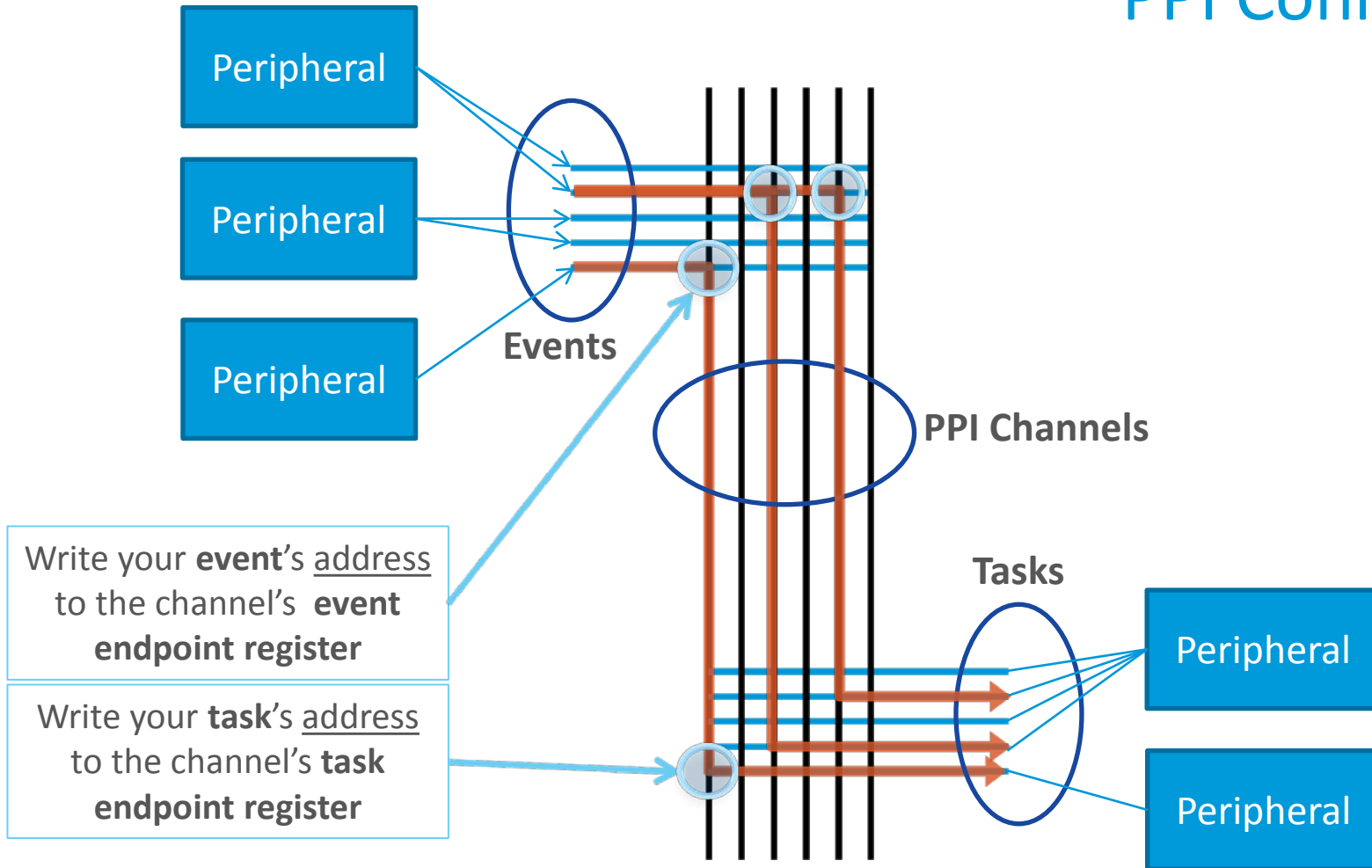
IRQ vs Programmable Peripheral Interconnect (PPI)



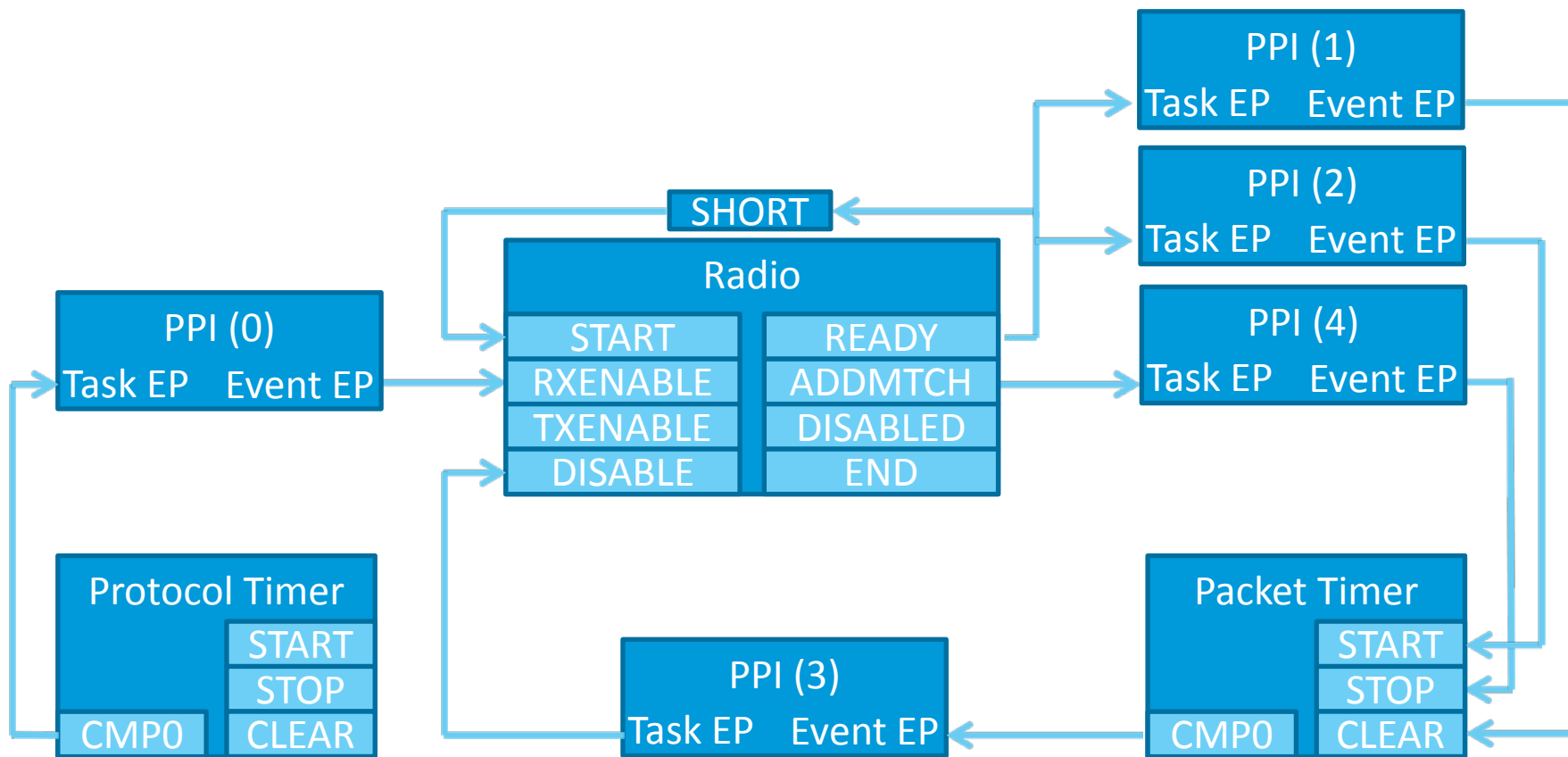
PPI - Block Diagram



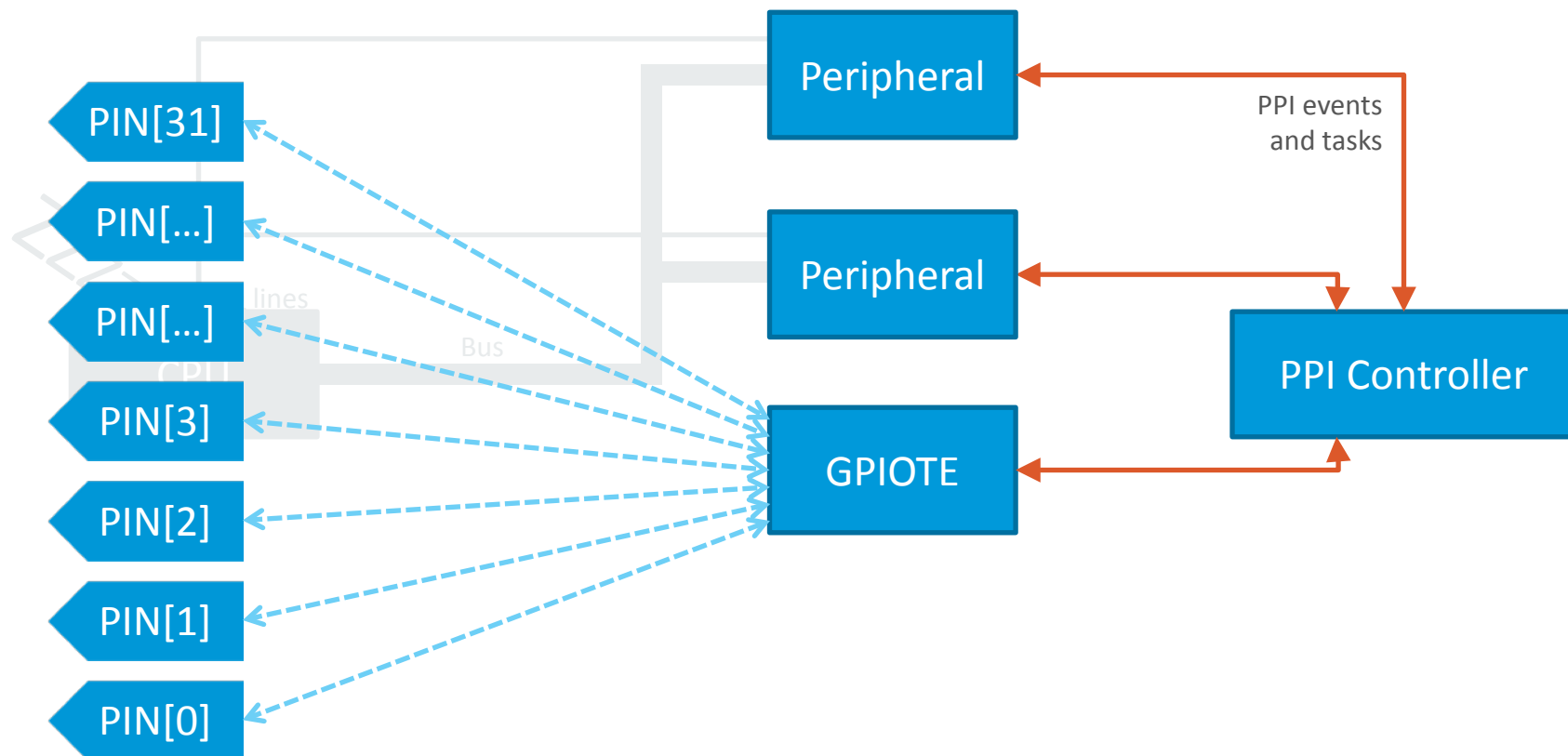
PPI Connections



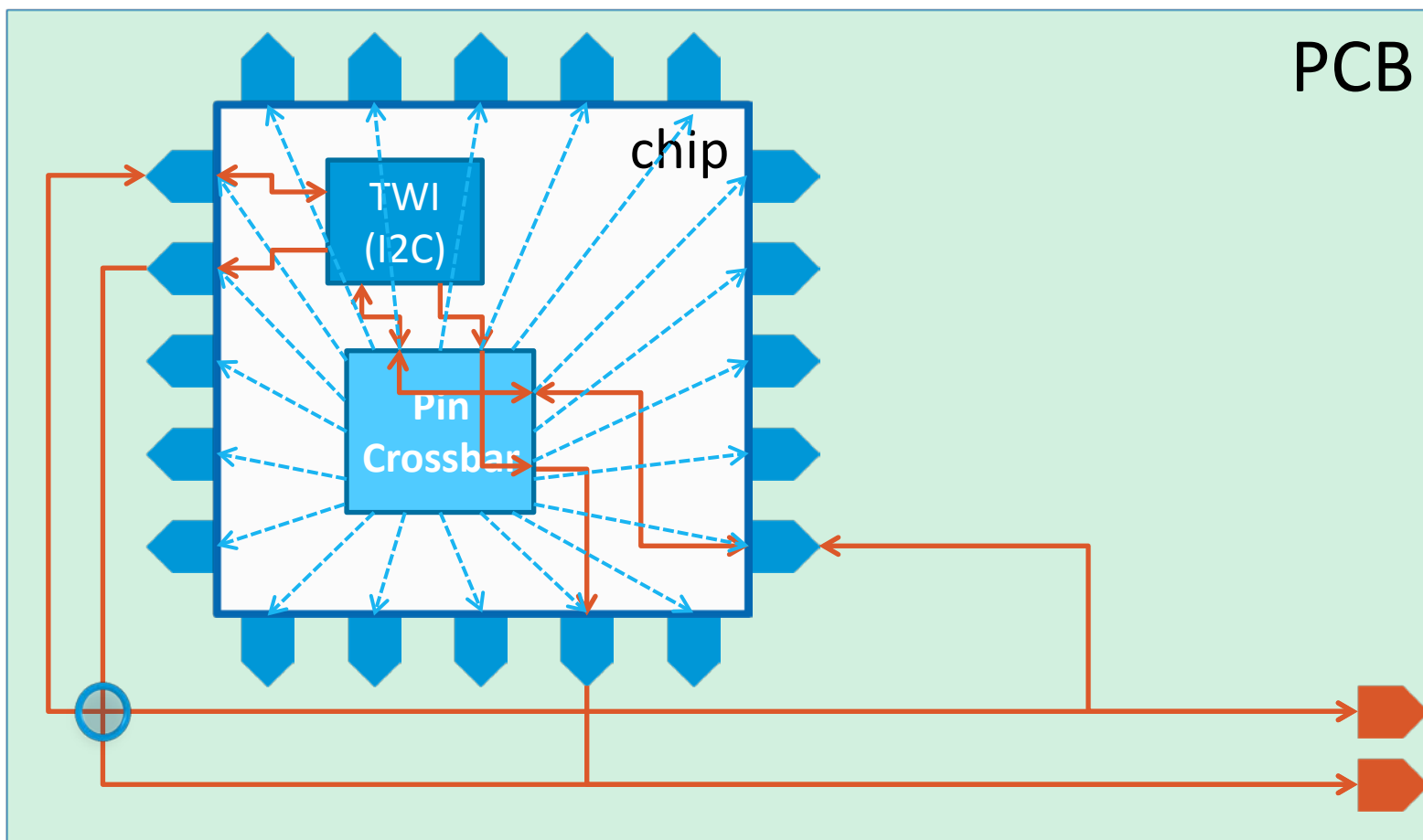
PPI - Real-life example



GPIOTE (GPIO Tasks and Events)



Peripheral pin access



Goal with nRF51 Series

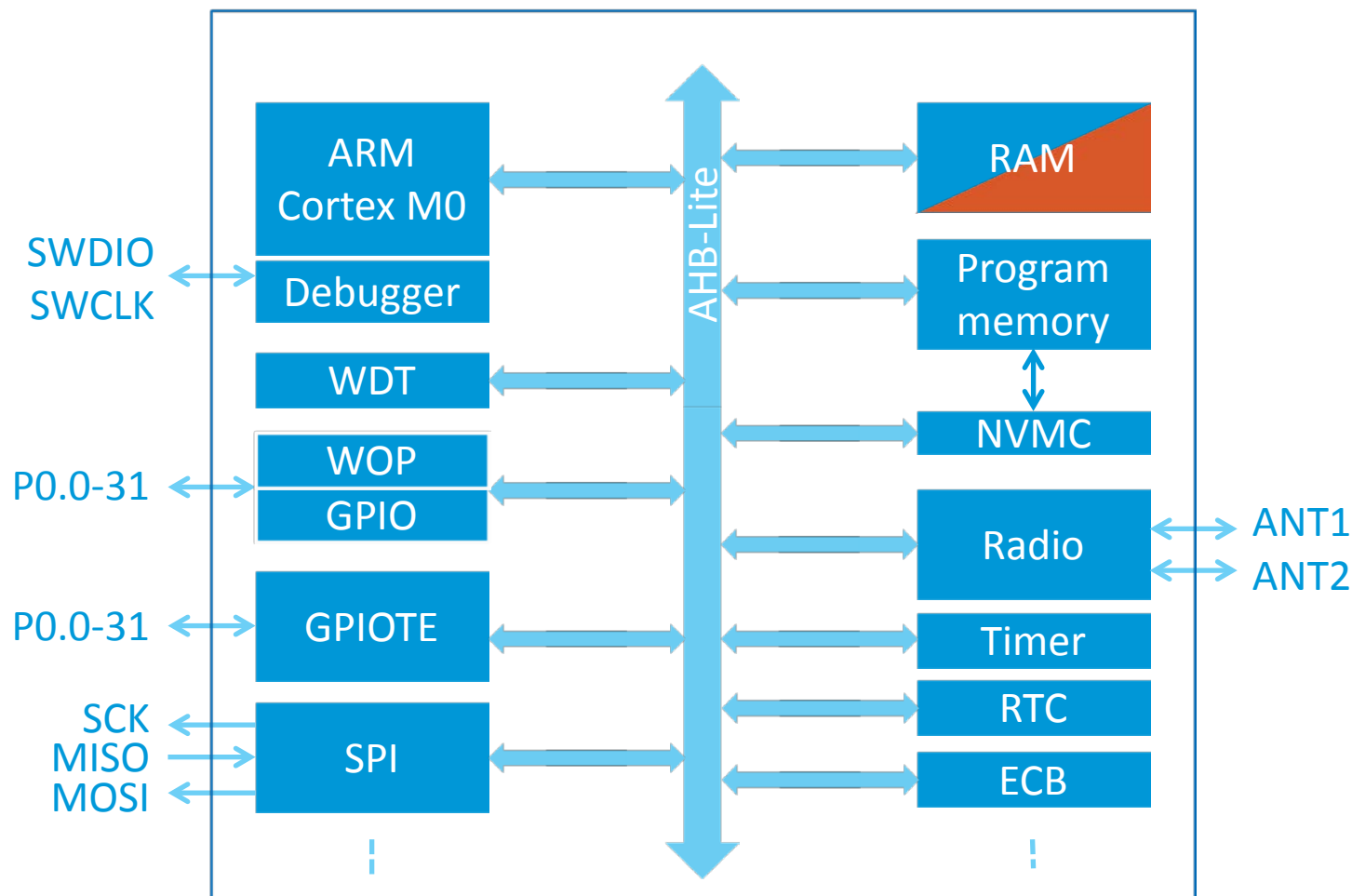
Flexibility

Consistency

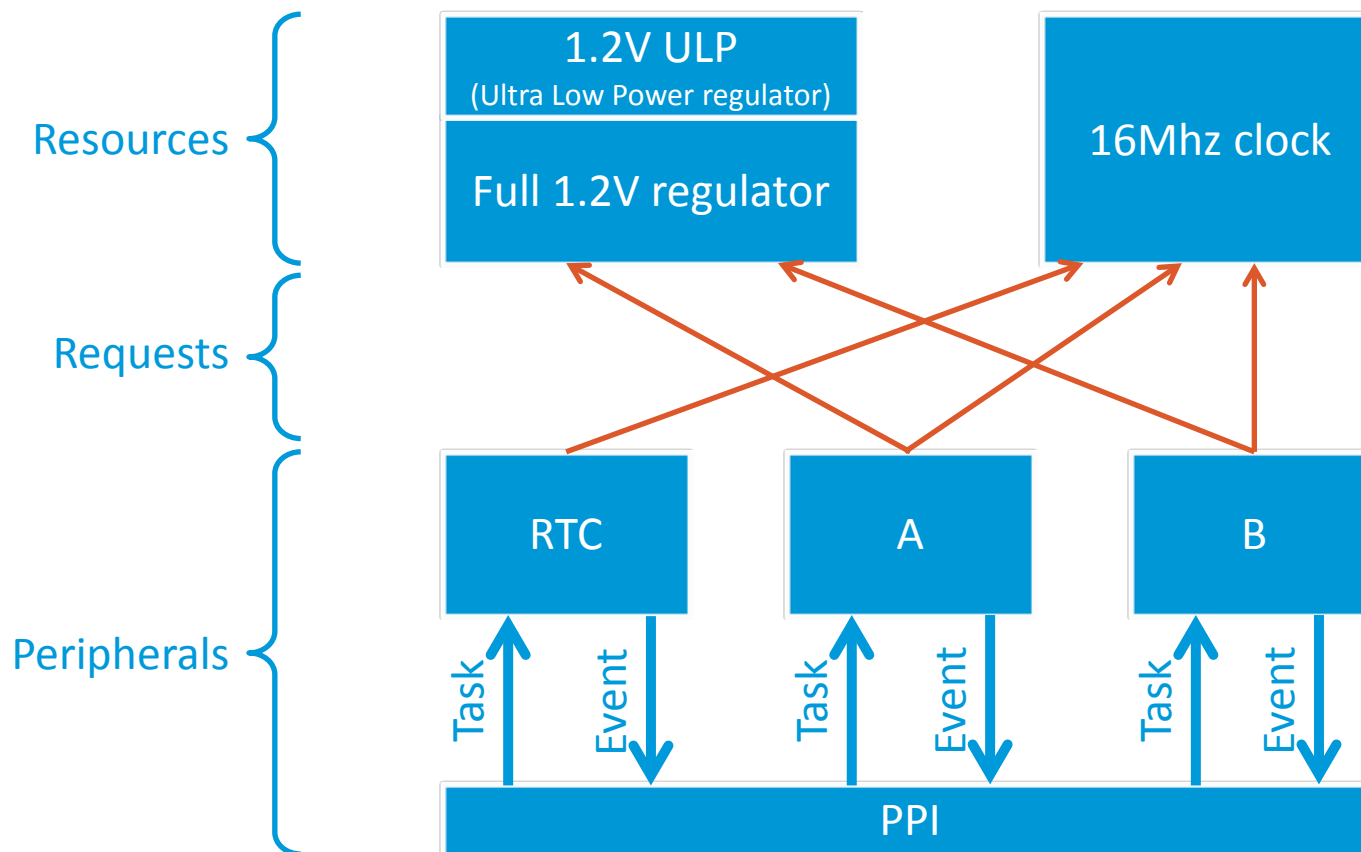
Low Power

Power and Clock

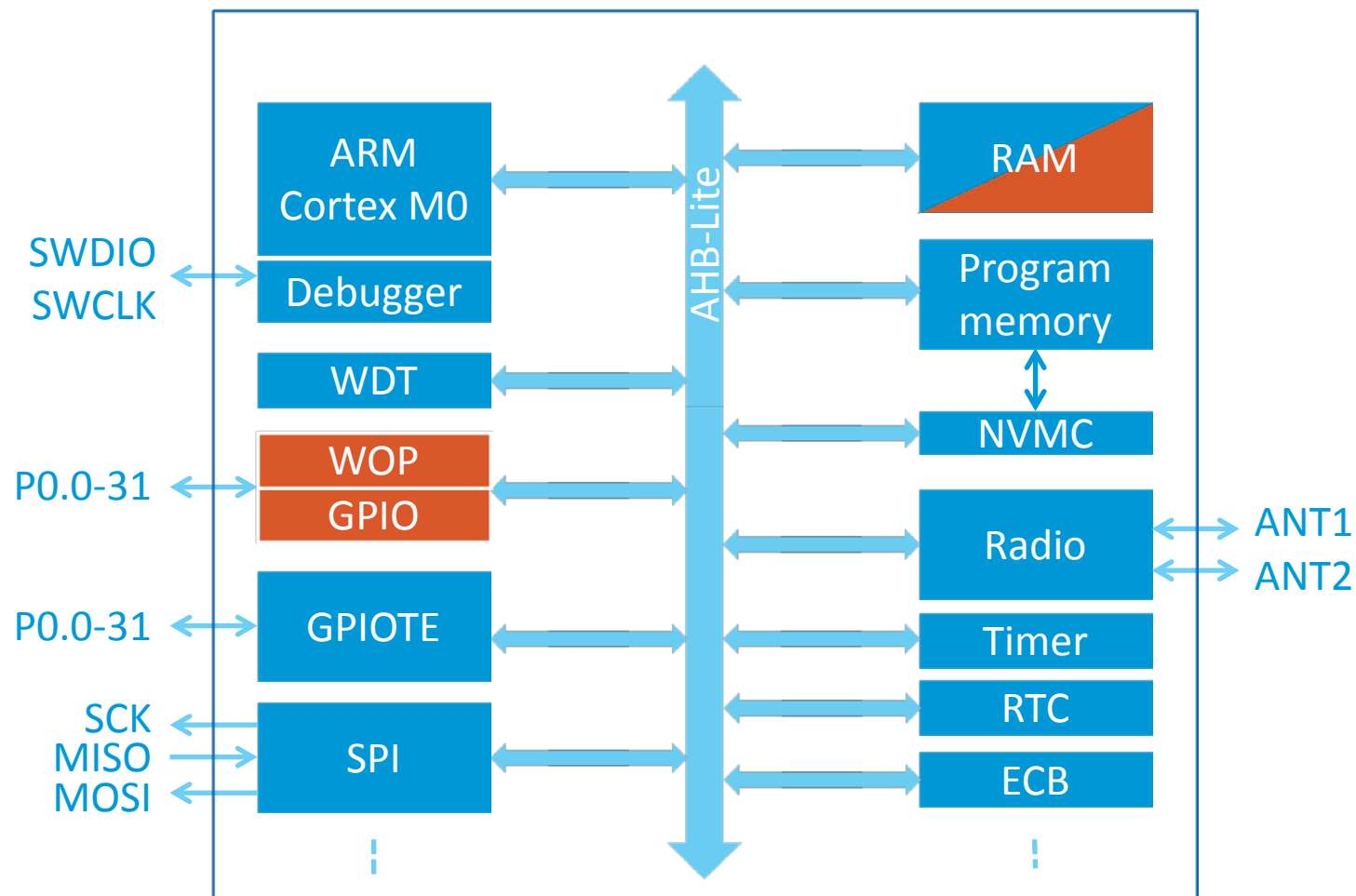
Power Management System Off



Power and Clock



System On



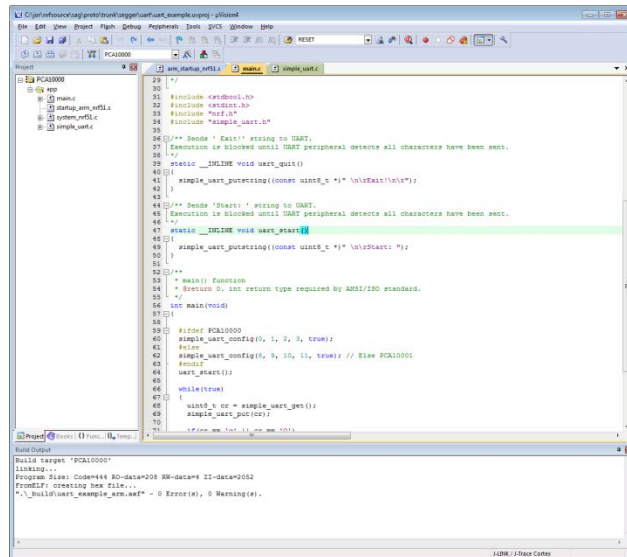
Goal with nRF51 Series

Flexibility

Consistency

Low Power

Code Reuse



```
29 //
30 //
31 #include <stdint.h>
32 #include <stdbool.h>
33 #include "nrf51.h"
34 #include "simple_uart.h"
35
36 /** Sends 'Exit!' string to UART.
37 Execution is blocked until UART peripheral detects all characters have been sent.
38 */
39 static __INLINE void uart_exit()
40 {
41     simple_uart_putstring((const uint8_t *) "Exit!\n");
42 }
43
44 /** Sends 'Start!' string to UART.
45 Execution is blocked until UART peripheral detects all characters have been sent.
46 */
47 static __INLINE void uart_start()
48 {
49     simple_uart_putstring((const uint8_t *) "Start!");
50 }
51
52 /**
53 * main() function
54 * Returns 0, the return type required by ARM/250 standard.
55 */
56 int main(void)
57 {
58     //
59     // PCAL0000
60     simple_uart_config(1, 2, 3, true);
61     //
62     simple_uart_init(9, 10, 11, true); // Same PCAL0000
63     //
64     uart_start();
65     while(true)
66     {
67         uint8_t c = simple_uart_get();
68         simple_uart_put(c);
69         //
70         // Print the int 11, the max 1011
71     }
72 }
```

Build Output

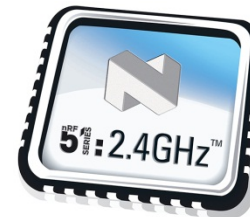
Build target 'PCAL0000'

Linking...

Program Size: Code=444 RO-data=208 RW-data=4 I2-data=2052

Printf: creating hex file...

~_build\uart_example_nrf51 - 0 Error(s), 0 Warning(s).



Goal with nRF51 Series

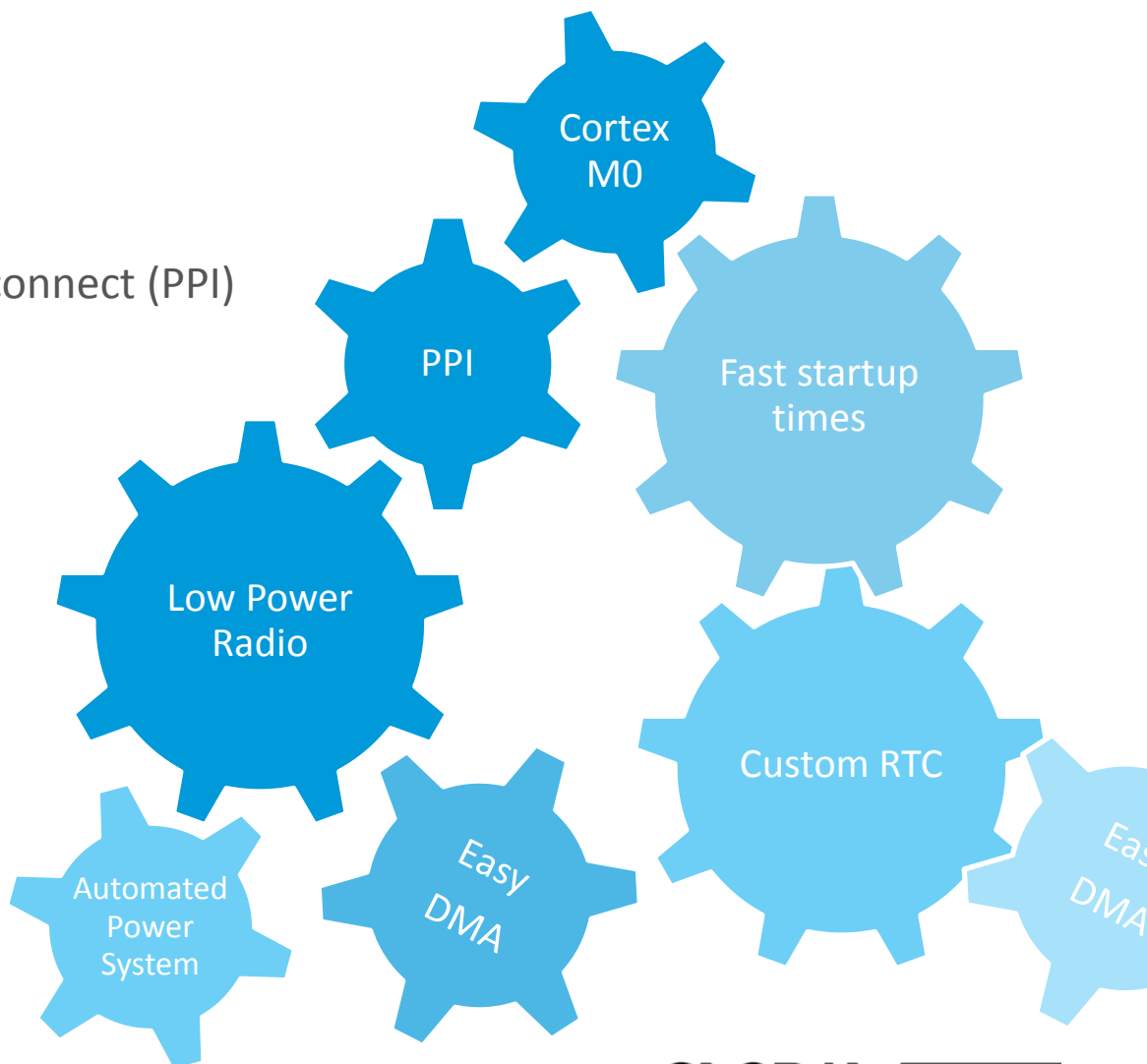
Flexibility

Consistency

Low Power

Ultra low power

- Programmable Peripheral Interconnect (PPI)
- Cortex M0
- Fast Startup Time
- Radio
- Automated Power System
- EasyDMA





nRF51 Series introduction

Thank you for your attention