ESN LAB 3 - 7 segment with avalon interface and IRQ Timer

note: for the best viewing experience, view this document in vscode with the markdown preview enhanced by Yiyi Wang extension installed, otherwise open the PDF version found here

to start cygwin from powershell:

```
& 'C:\intelFPGA\18.1\nios2eds\Nios II Command Shell.bat'
```

to generate bsp:

```
nios2-bsp hal ./software/bsp/ ./*.sopcinfo
```

to generate makefile:

```
nios2-app-generate-makefile --app-dir ./software/app --bsp-dir ./software/bsp --elf-name
```

both (one liner):

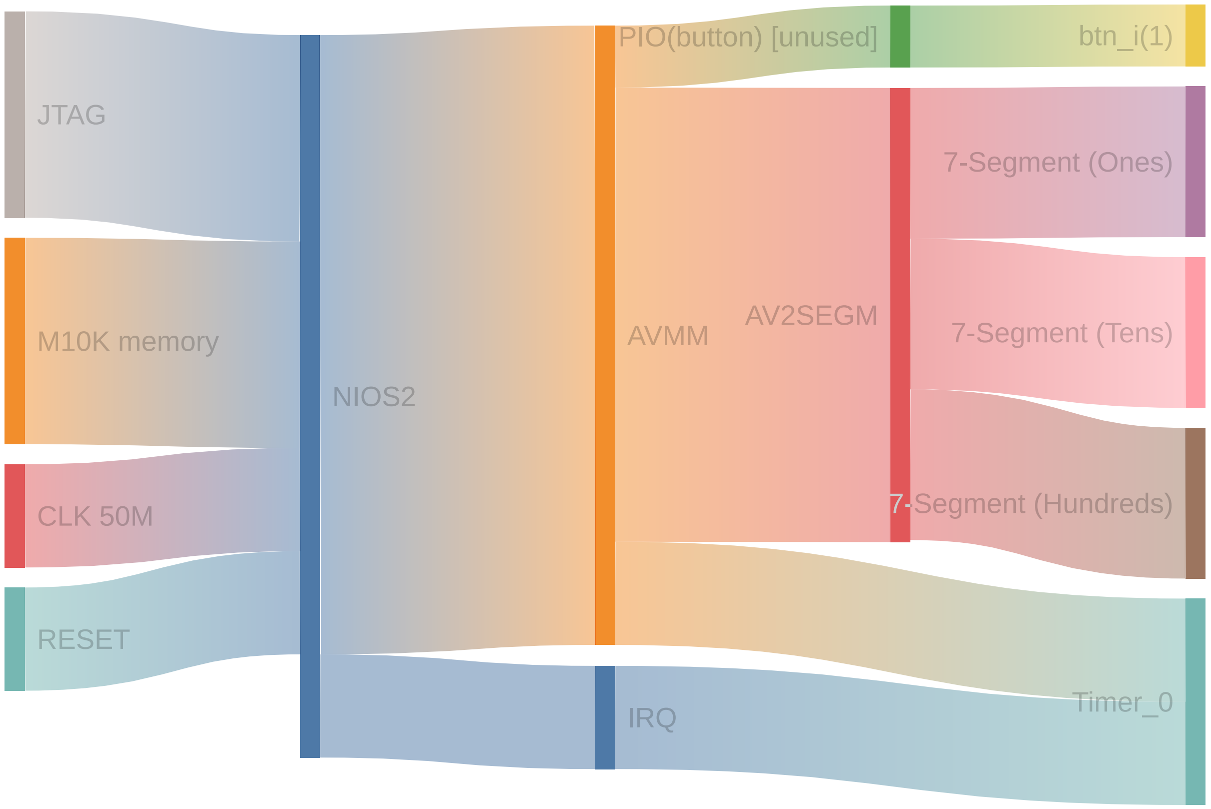
```
nios2-bsp hal ./software/bsp/ ./*.sopcinfo && nios2-app-generate-makefile --app-dir ./so
```

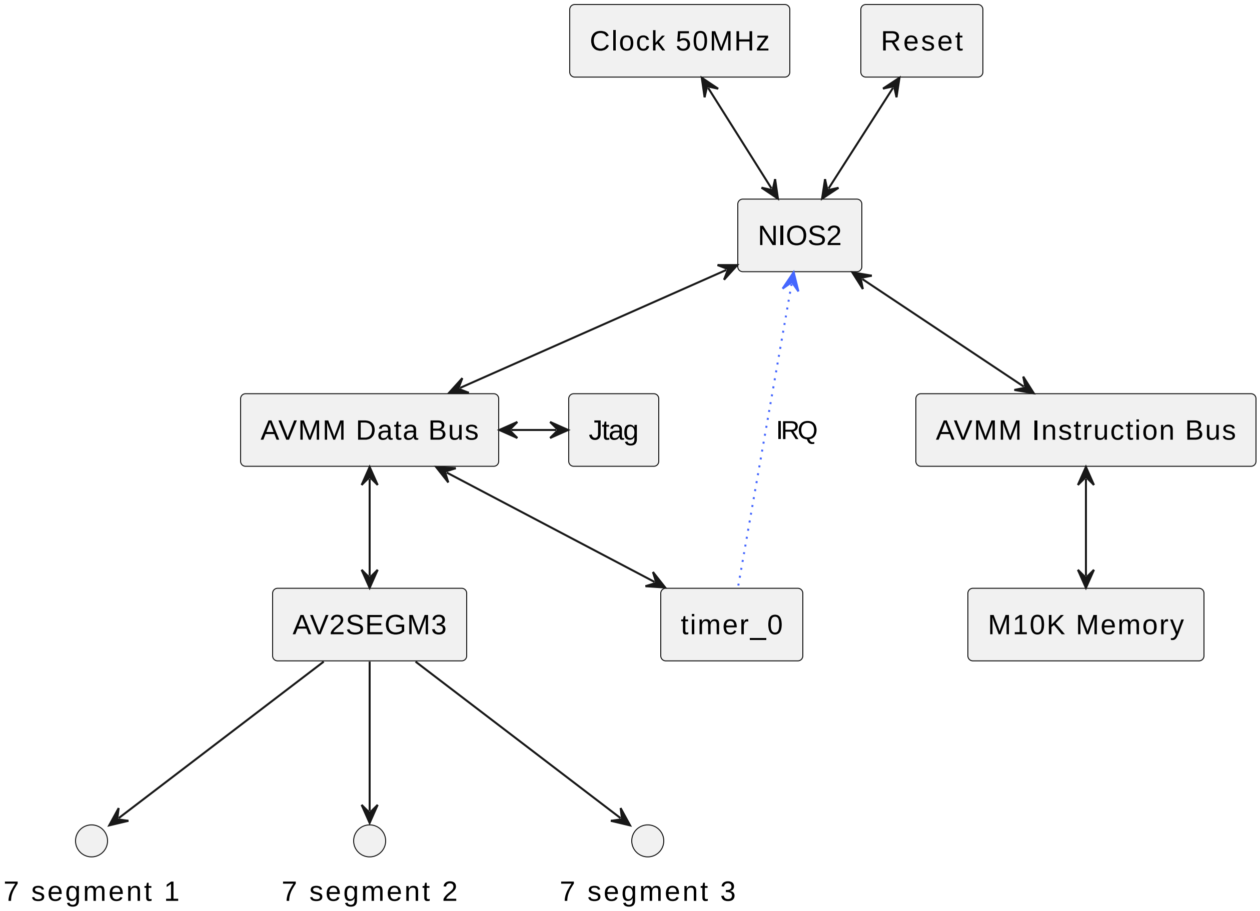
Introduction

In this lab we have to add a timer IP to the SoPc so that the NISO2 cpu gets one IRQ per second to precisely count up.

System architecture

This will be quite similar to LAB2, with an added Timer connected to AVMM for setup and with an IRQ to report the tick.


JTAG
M10K memory
CLK 50M
RESET
NIOS2
PIO(button) [unused]
AVMM
AV2SEGM
IRQ
btn_i(1)
7-Segment (Ones)
7-Segment (Tens)
7-Segment (Hundreds)
Timer_0


Clock 50MHz
Reset
NIOS2
AVMM Data Bus
Jtag
IRQ
AVMM Instruction Bus
AV2SEGM3
timer_0
M10K Memory
7 segment 1
7 segment 2
7 segment 3

Progress

The system is functionnal, both the 1 and 3 7 segment counters have been implmented, backed by C logic from the NIOS over the AVMM bus.

Conclusion

The AVMM bus widths need to be managed with care as using an 8 bit bus width will cause the default 32 bit IOWR/RD functions from the NIOS2 HAL to fail, to fix this using the 8 bit variants of these functions is require, as follows:

```
IOWR_8DIRECT(AV2SEGM3_0_BASE, 0x0, ones);
alt_printf("ones: %x\r\n", ones);
IOWR_8DIRECT(AV2SEGM3_0_BASE, 0x1, tens);
alt_printf("tens: %x\r\n", tens);
IOWR_8DIRECT(AV2SEGM3_0_BASE, 0x2, hundreds);
alt_printf("hundreds: %x\r\n", hundreds);
```

The actual AV2SEGM3 ip (the one with 3 7 segments) has been implemented using an address mode set of AVMM registers, where each register corresponds to one 7 segment.