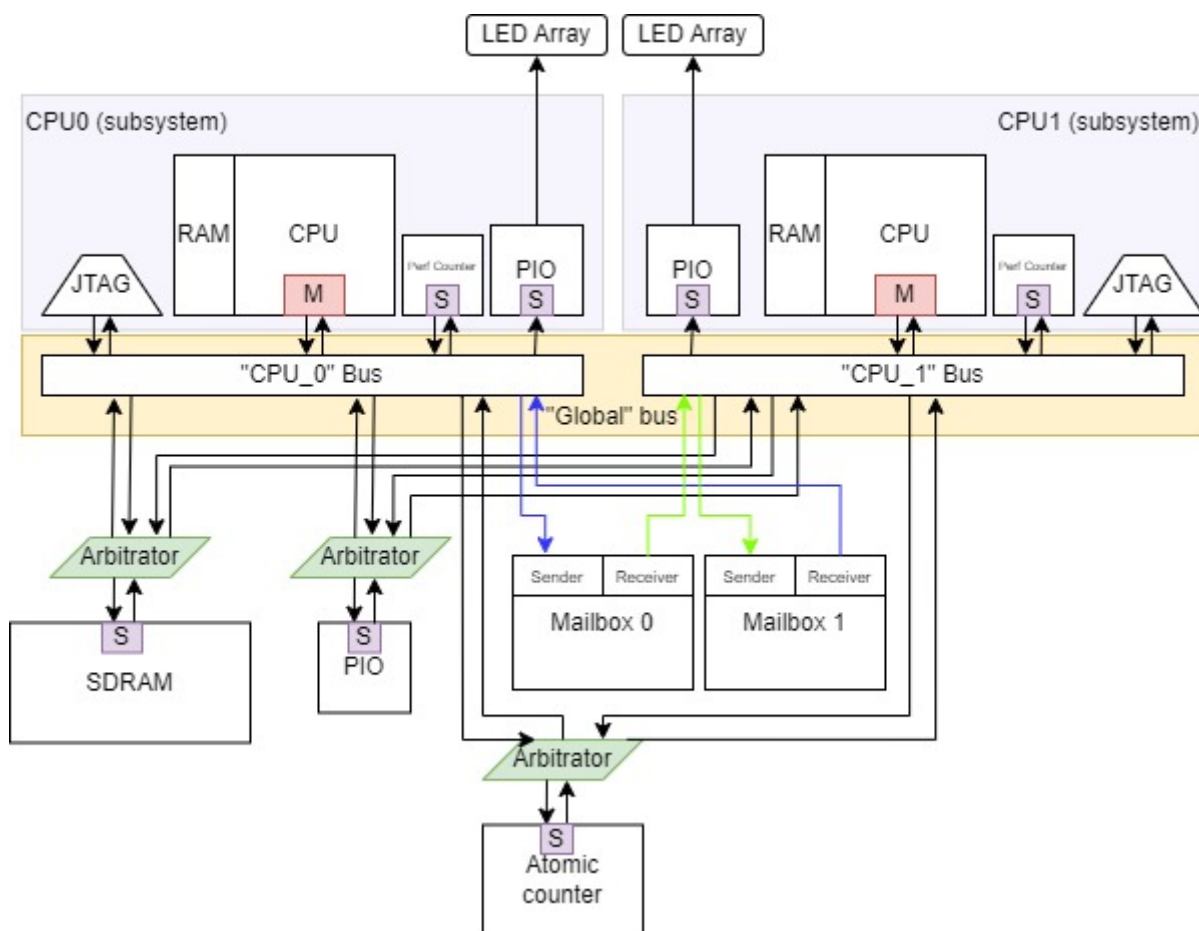


# Lab3 - Multiprocessors

PELUSO Nathan - KERJAN Tugdual

## Design

### System diagram



### Provided files

```
.
├── hw
│   ├── hdl
│   │   ├── atomic_counter.vhd           # VHDL file for custom "atomic
│   │   │   increment" counter
│   │   └── DE1_SoC_top_level.vhd       # Top-level file for multi-CPU
design
│   └── quartus
│       ├── soc_system.qsys             # qsys file (including
subSystems for both CPUs)
│       └── subsystemA                  # Quartus-generated subsystem
folder
│   └── ...
```

```

└─ sw                                     # NIOSII software files
    └─ lab3_cpu0                         # C files to use for CPU_0
        └─ atomic_counter.c
        └─ parport_mailbox.c
        └─ parport_mutex.c
        └─ parport_test.c
    └─ lab3_cpu1                         # C files to use for CPU_1
        └─ atomic_counter.c
        └─ parport_mailbox.c
        └─ parport_mutex.c
        └─ parport_test.c

```

## Manipulations

---

### Parallel port test

#### Files to use

- parport\_test.c (CPU\_0 & CPU\_1)

#### Explanation

Simple software designed to test separated PIO (by blinking independent LEDs) and try concurrently increasing a value by accessing an 'open-ended', shared PIO.

### Hardware Mutex

#### Files to use

- parport\_mutex.c (CPU\_0 & CPU\_1)

#### Explanation

Make access to the PIO "counter" exclusive by using a Hardware Mutex component. Measure performance to increase this counter.

#### Statistics

##### CPU0 (incrementing every 20ms)

```

--Performance Counter Report--
Total Time: 9.99979 seconds (499989530 clock-cycles)
+-----+-----+-----+-----+-----+
| Section      | %   | Time (sec)| Time (clocks)|Occurrences|
+-----+-----+-----+-----+-----+
|Counter incr. |0.0741| 0.00741|      370605|        500|
+-----+-----+-----+-----+-----+

```

CPU1 (decrementing every 20ms)

```
--Performance Counter Report--
Total Time: 5.00981 seconds (250490362 clock-cycles)
+-----+-----+-----+-----+-----+
| Section          | %    | Time (sec)| Time (clocks)|Occurrences|
+-----+-----+-----+-----+-----+
| Counter incr.    | 0.13 | 0.00653   | 326646       | 500       |
+-----+-----+-----+-----+-----+
```

Hardware Mailbox

Files to use

- parport\_mailbox.c (CPU\_0 & CPU\_1)

Explanation

Use Hardware Mailbox components (only one is currently used but two are available to allow back-and-forth communication) to make a process from one CPU send a message to a process of the other CPU. To limit the size of the message sent and allow for easy arbitrary data transfers, the data is stored on shared SDRAM and only the adress of the stored data (and its size) is sent from one CPU to the other.

Atomic Counter

Files to use

- atomic\_counter.c (CPU\_0 & CPU\_1)

Explanation

Create a custom "Atomic counter" component in VHDL, allowing the following operations:

- Initialize counter to given value (default: 0)
- Increment counter (by writing a single bit)
- Read counter value

This component is shared between the two CPUs.

Results

The designed counter has atomic write: a write pulse atomically increases the counter value by one. The read operation is untouched, and a pair "write-read" or "read-write" is still not atomic: if both processors increase the counter value then read, one of them could see a +2 increase to prevent this, using the mutex is still necessary. The mutex could also be necessary to avoid "reset-increase" conflicts.

Statistics

CPU0

```
--Performance Counter Report--
Total Time: 9.9998 seconds  (499990205 clock-cycles)
+-----+-----+-----+-----+-----+
| Section          | %   | Time (sec)| Time (clocks)|Occurrences|
+-----+-----+-----+-----+-----+
|Counter incr.    |0.0008| 0.00008|          4000|          500|
+-----+-----+-----+-----+-----+
```

CPU1

```
--Performance Counter Report--
Total Time: 9.9998 seconds  (499990095 clock-cycles)
+-----+-----+-----+-----+-----+
| Section          | %   | Time (sec)| Time (clocks)|Occurrences|
+-----+-----+-----+-----+-----+
|Counter incr.    |0.0008| 0.00008|          4000|          500|
+-----+-----+-----+-----+-----+
```

The performance is significantly better without the mutex.