label22Captionfigure.caption.3 _ label@creflabel@cref[figure][2][]2[1][2][]2

Points	Grade

Team: XX

1326907 David FREISMUTH

Digital Integrated Circuits Lab (LDIS)

384.088, Summer Term 2019

Supervisors:
Christian Krieg, David Radakovits, Axel Jantsch

Task 2:
Design Characterization, Bus Communication

1 Characterize your design from Task 1

In order to make your design implementation comparable to other implementations, simulate your design using Vivado and perform the following measurements:

1. Timing analysis
2. Power analysis
3. Resource consumption

Create a design space vector for your design.The design space vector holds the following parameters, where t_{max} is the maximum delay for the critical path, P_{avg} is the average power consumption for your design (vector-less post-place-and-route power estimation), and r is the percentage of resources used in your implementation for the Nexys 4 DDR board (for sake of simplicity, use the percentage of slices):

$$\vec{v} = \begin{bmatrix} t_{max} \\ P_{avg} \\ r \end{bmatrix} \tag{1}$$

You will find information on power estimation[1] and timing analysis[2] on the web.

Get the design space vectors of your colleagues, and visualize the three-dimensional design space. Use black crosses for the vectors of your colleagues, and a filled circle for your own vector.

2 Create an AMBA APB interface

The goal of this subtask is to make your design accessible from other intellectual property (IP) cores via the advanced microcontroller bus architecture (AMBA) advanced peripheral bus (APB). Consult the APB specification,[3] and:

1. Implement a bus controller that implements the use case for your task
2. Implement a bus interface for any sub-component of your design from Task 1 (sampling, data processing, output).
3. Implement a module that takes the user input for runtime parameters and connect it to the bus

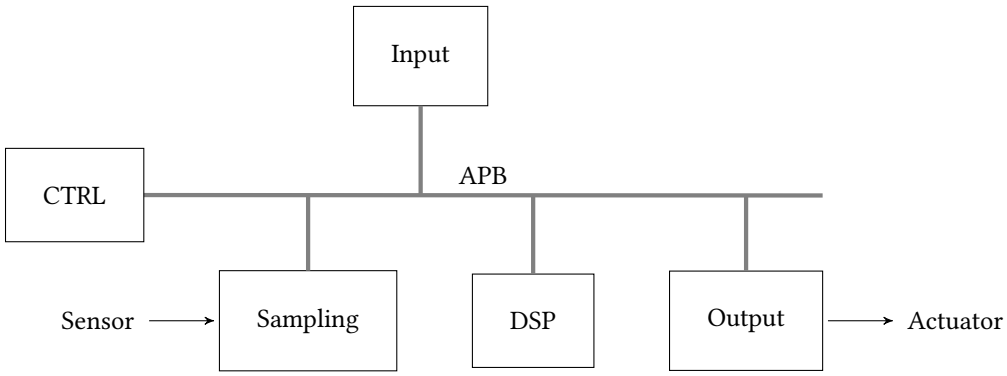

Figure 1: System of Task 1 equipped with an APB

Your system should be able to take the user input, parametrize the system according to the user input, and perform the actual task. Instead of directly connecting each of the sub-components directly, the modules should communicate over the APB. The audio system can be treated as one module; as it is kind of a real-time system, it wouldn't make a lot of sense to let it communicate over the APB. The user-input (e.g., cut-off freuqency) should be set over the APB.

[1] **xpe**.
[2] **timing**.
[3] **apb**.

3 Task 1

In this task, the metric introduced in Section 1 shall be applied on the design, that has been established in the previous task. The values that are used to calculate the vector, can be found in Vivado.

Critical Path Delay t_{max}

After successful implementation, the delay of the critical path t_{max} can be retrieved by executing the command `report_timing_summary` in the tcl console. The value `DataPathDelay` describes the critical path delay.

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             90.518ns  (required time - arrival time)
  Source:                 ldis_task1_bd_i/ADXL362Ctrl_0/U0/ACCEL_Y_reg[11]/C
                            (rising edge-triggered cell FDRE clocked by clk_out1_ldis_task1_bd_clk_wiz_0_0  {rise@0.000ns fall@50.000ns period=100.000ns})
  Destination:            ldis_task1_bd_i/quadratic_func_0/U0/x2_a_reg/B[8]
                            (rising edge-triggered cell DSP48E1 clocked by clk_out1_ldis_task1_bd_clk_wiz_0_0  {rise@0.000ns fall@50.000ns period=100.000ns})
  Path Group:             clk_out1_ldis_task1_bd_clk_wiz_0_0
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            100.000ns  (clk_out1_ldis_task1_bd_clk_wiz_0_0 rise@100.000ns - clk_out1_ldis_task1_bd_clk_wiz_0_0 rise@0.000ns)
  Data Path Delay:        8.482ns  (logic 6.073ns (71.602%)  route 2.409ns (28.398%))
  Logic Levels:           9  (CARRY4=7 DSP48E1=1 LUT2=1)
  Clock Path Skew:        -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    -3.654ns = ( 96.346 - 100.000 )
    Source Clock Delay      (SCD):    -3.800ns
    Clock Pessimism Removal (CPR):    -0.291ns
  Clock Uncertainty:      0.226ns  ((TSJ^2 + DJ^2)^1/2) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Discrete Jitter          (DJ):    0.447ns
    Phase Error              (PE):    0.000ns
```

Figure 2: tcl console output, after executing the command `report_timing_summary`.

Average Power Consumption P_{avg}

The average power consumption P_{avg} can be seen in the project summary view. See **??** for reference.

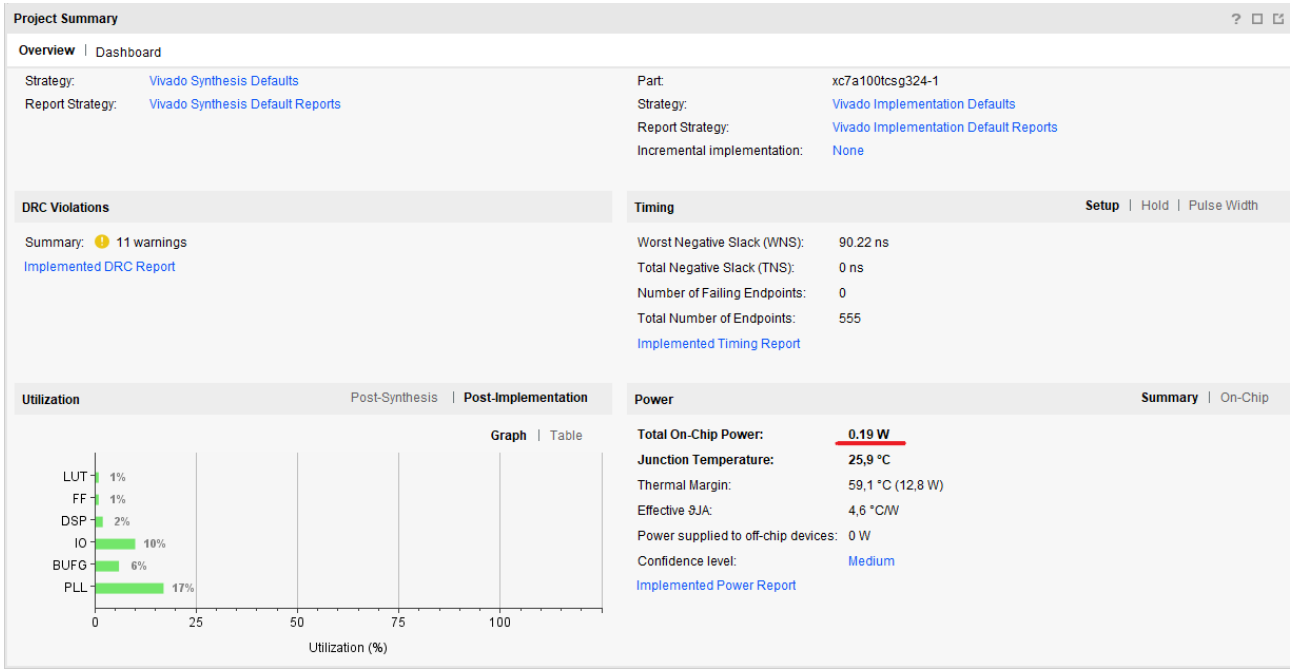

Figure 3: The project summary view of a Vivado project. The average power consumption value has been marked.

Resource Consumption r

The resource consumption shall be measured by the usage of logical slices. It is calculated from the values retrieved by the tcl command`report_utilization`. **??** shows the formula that is used to compute the resource consumption.

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	693	0	63400	1.09
LUT as Logic	693	0	63400	1.09
LUT as Memory	0	0	19000	0.00
Slice Registers	498	0	126800	0.39
Register as Flip Flop	498	0	126800	0.39
Register as Latch	0	0	126800	0.00
F7 Muxes	0	0	31700	0.00
F8 Muxes	0	0	15850	0.00

Figure 4: tcl console output, after executing the command `report_utilization`.

$$\frac{\text{Slice Luts used} + \text{Slice Registers used} + \text{F7 Muxes used} + \text{F8 Muxes used}}{\text{Slice Luts available} + \text{Slice Registers available} + \text{F7 Muxes available} + \text{F8 Muxes available}} \tag{2}$$

Results

By applying the metrics established in **??**, the vector in **??** is yielded.

$$\vec{v} = \begin{bmatrix} t_{max} \\ P_{avg} \\ r \end{bmatrix} = \begin{bmatrix} 8,896ns \\ 0,190W \\ 0,005 \end{bmatrix} \tag{3}$$

The design space vectors of other solutions can be seen in **??**. **??** visualizes those vectors.

t_{max} [ns]	P_{avg} [W]	r [%]
8,896	0,190	0,005
2,000	0,111	0,150
3,000	0,207	1,000

Figure 5: Design space vector of other solutions.

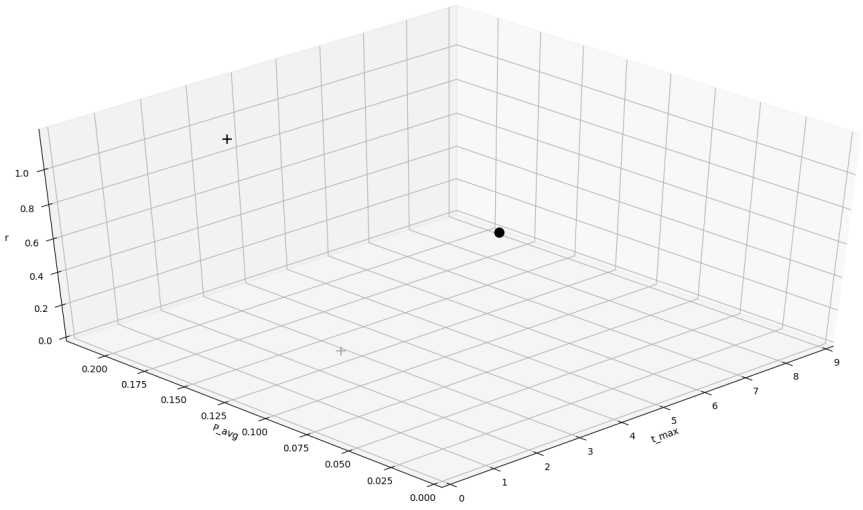

Figure 6: The plot of all design space vectors. Crosses mark the vectors of other designs. The circle marks the vector of the design that has been established during the course of this task.