1 PURPOSE

The Chip (Integrated Circuit) Design Competition, which was opened to university students for the first time in 2022 within the scope of TEKNOFEST, aims to raise awareness in the field of microelectronic technologies in associate, undergraduate and graduate (master) students and to provide students with knowledge on IC design. The competition is also organized to encourage teamwork and to lead the formation of competent human resources in this field by developing micro-electronic technologies with practical applications.

This document has been created to define all the rules and requirements of the Chip Design Competition, which is organized by the TÜBİTAK BİLGEM Integrated Design and Education Laboratory (TÜTEL) and Yongatek Microelectronics within the scope of TEKNOFEST Aviation, Space and Technology Festival (TEKNOFEST) Technology Competitions. The analog and digital processor design categories are organized by **TUTEL**, the microcontroller design category is organized by **Yongatek Microelectronics**.

2 COMPETITION SCHEDULE

Table 1: Competition Schedule

DESCRIPTION	DATE
Application Deadline for the Competition	20.02.2024
Submission Deadline for Preliminary Design Report (PDR)	15.03.2024 17:00
Announcement of the Teams Passing the Preselection based on PDR Results	22.03.2024
QA Session	15.04.2024
Submission Deadline for Detail Design Report (DDR)	15.06.2024 17:00
Announcement of Finalist Teams	01.07.2024
QA Session	08.07.2024
Deadline for Finalizing the Design	31.07.2024
Final Presentations and Demos	Not yet determined

3 GENERAL INFORMATION ABOUT THE COMPETITION

The competition consists of three categories; analog, digital processor, and microcontroller. For the competitions organized by TÜTEL, it is expected that a customized processor with RISC-V Instruction Set Architecture (ISA) will be designed in the digital processor design category. Designing a bandgap reference circuit, a LDO voltage controller, and a Voltage Controlled Oscillator (VCO) is expected in the analog design category. In the microcontroller design category organized by Yongatek Mikroelektronik, it is expected to design and verify a microcontroller with various interface elements and memory units using a ready-made processor core.

What is expected from the teams that will participate in the competition is to prepare the preliminary design report, detailed design report, complete the design, prepare the design outputs and make a presentation/demo in the final for the hardware that meets the design criteria specified in this document.

Applications will be made through the official website of TEKNOFEST Technology Competitions (www.teknofest.org) until February 20, 2024.

3.1 Participation Rules

- Higher education (associate, undergraduate and master) students studying in Turkiye and abroad can participate in the competition.
- Participation in the competition can be made individually or as a team. Teams must consist of a maximum of 5 people (excluding consultant).
- Number of graduate students (master) in the teams cannot exceed 2.
- A competitor can take part in teams in different categories, but not in different teams in the same category.
- Teams can only have 1 person as a consultant. It is not required to have a consultant. A consultant can advise only one team per category.
- Along with the competition application, by using KYS system, approved student documents must be submitted for students, and a certified document showing that they are a lecturer/staff or research assistant must be submitted for advisors.
- Teams can be formed from a single school or as a mixed team with one or more higher education students coming together.
- Each team participating in the competition can submit only one design.
- Competitors who have completed the team formation process must apply to the competition in accordance with their project.
- Between the application dates, the team captain/advisor registers through the system, makes the correct and complete registration of the consultant and/or team captain/team members, if any, and sends an invitation to the advisor and members' e-mails, if any. The member to whom the invitation is sent, logs in to the Application system, accepts the invitation from the "My Team Information" section and the registration is completed. Otherwise, the registration will not be completed.
- All necessary processes within the scope of the competition (Application, Report Submission, Announcement of the Report Results, Financial Support Application, Objection, Member Addition/Removal, etc.) are done through the KYS system. Teams are required to follow their processes through the KYS system.
- Adding/removing members can be done until the Detail Design Report Submission date.
- During the competition process, the processes of applying through the KYS, uploading reports and filling out forms are under the authority of the team captain and/or the consultant, and the competition processes are managed through these people.

- Transportation and accommodation support to be provided to the finalist teams is limited. The number of people to be supported is 3 people (including the consultant) per team and TEKNOFEST Competitions Committee has the right to make changes.
- TEKNOFEST Competitions Committee has the authority to limit the number of members in the festival area. In case of restrictions, the committee will inform.
- Throughout the competition process, education level at the time of application will be considered.
- Applications of TÜTEL and Yongatek Microelektronics employees will not be evaluated.

3.2 Reports and Presentation

- The format in the GENERAL RULES section should be followed for references from previous year reports. Other than that, IEEE format should be followed for references from other sources.
- Two reports, a preliminary design report and a detail design report, will be submitted. The technical information required in the relevant headings of the reports should be conveyed in detail.
- Report templates are published on the competition website.
- Reports can be prepared in Turkish or English. It is mandatory to use a single language in a report.
- The reports are uploaded to the application system in PDF format with a maximum size of 60 MB until 05.00 pm in the specified deadlines. Otherwise, the team will be eliminated from the competition.
- The reports should be prepared in A4 format, 11 point, Calibri font, with a line spacing of 1.15 and 2.5 cm at the bottom, top and sides. The preliminary design report should have a maximum of 9 pages, and the detailed design report should have a maximum of 30 pages (including the cover page, pictures, tables, references).
- The report is evaluated within the scope of scientific and technical criteria specified by DDK members who are experts in their fields. Teams with a score below what is determined by DDK are eliminated from the race.
- Teams with successful reports are announced on the www.teknofest.org page.
- In case of an ethical violation such as plagiarism, copying etc., the report will not be evaluated and the corresponding team will be eliminated from the competition. This rule is applied for the application of the same university/club/society etc. regardless of category/race.

3.2.1 Preliminary Design Report (PDR)

Preliminary design report is expected to include the following sections:

Block diagram of the overall design, target performance summary table, system level importance of targeted performance measures, schematics of the possible critical circuit core, verification plan and methods, brief description of the circuit and design

techniques to be used, applicable references (e.g., an article that inspires your design, all the open-source blocks you'll reuse, etc.). The design structure given in the PDR can be changed later, provided that change reasons are explained.

The up-to-date report template will be shared on the competition website.

In order to proceed to the next stage within the scope of this competition, the preliminary design report must be submitted and deemed successful.

3.2.2 Detailed Design Report (DDR)

In the detailed design report of the digital processor design category, the details of the design, how the problems presented in the specification were handled, block diagrams, simulation and IC flow results should be included.

The up-to-date report template will be shared on the competition website.

Teams that have passed to the DDR stage are obliged to submit their reports on the date specified in Table 1.

According to the DDR results, the teams that will participate in the final evaluation will be announced on the date specified in Table 1.

3.2.3 Final Evaluation Presentation and Design Outputs

Final evaluation presentation and design outputs (schematic, RTL, GDSII etc.) must be submitted by the date specified as “Deadline for Finalizing the Design” in Table 1. Changes made to GitHub repositories after 11.59 pm on the relevant date will not be taken into consideration. The presentations should be uploaded on the same date until 05.00 pm. Presentations in the competition area will be made through presentation files uploaded to the system. Using the design outputs, the results in the presentation will be verified and the non-conforming groups will be eliminated. For this reason, the design outputs must be up-to-date and compatible with the presentation. Details of the design outputs are given in Design Outputs section.

3.3 Evaluation

Teams that pass the PDR stage will be entitled to participate in the DDR stage. PDR and DDR reports will be evaluated in accordance with the template to be announced

on the competition website. Teams that pass the DDR stage will be eligible to participate in the presentation and demo evaluation. Presentation evaluation will be made by the competition jury according to the oral presentation and the answers to the questions of the evaluators. Teams that are not in the competition area although they are entitled to participate in the presentation evaluation are considered as withdrawn from the competition. First, second and third competitors will be determined according to the weighted points of all stages.

3.3.1 Digital Processor Design Category

Evaluation consists of several stages. Firstly, the processor (RV32IMAFB_Zicsr (only the machine mode) instruction set processor, UART peripheral) requested from the competitors will be put into a test environment, and the interface of the processor should be compatible with the wrapper module that will be shared with the competitors. Test codes will be loaded into the main memory located in the top module wrapper. In this test environment, it will be tested whether or not the competitor implements the instruction set correctly. During the tests, some boundary conditions (such as dividing a number by zero) will also be tested. For testing the UART block, commands will be sent to this block over the peripheral bus in accordance with the register address space given in APP-3. Secondly, this block will be tested with the processor first in the simulation environment, then on the FPGA.

Projects that have been tested and found to meet minimum requirements will be subjected to performance analysis. In this analysis, various performance tests (e.g., CoreMark) will be run on the processor. IC flow, design evaluation, and verification work will be carried out on the designs after presentations and demos. The total score that can be obtained at the end of the competition will be a maximum of 100 points and the calculation will be made as follows.

Total Points = (0.1 * PDR Points) + (0.2 * DDR Points) + (0.2 * Presentation Points) + (0.1 * Demo Points) + (0.25 * IC Flow Points) + (0.15 * Design and Verification Points)

Expectations and criteria in scoring will be shared later. Those who have not updated their GitHub repo in at least one of the 20-day intervals following the later-announced release date will lose points from the Design and Verification evaluation. In case of a tie, the team with the most Chip Flow plus Design and Verification points will advance in the ranking.

3.4 Contact

For technical questions about the contest, communication will be made through the messaging group to be determined later. It is the responsibility of the competing team to actively follow this group and to follow the announcements and questions & answers in this group. Referees and jury committees are not responsible for the failure of the teams to reach up-to-date information as a result of not following the specified e-mail group.

Questions about the organizational parts of the competition should be submitted via iletisim@teknofest.org e-mail address.

It is important that your technical and organizational questions are conveyed through the correct channels above, in order to be able to respond quickly to the questions asked.

4 TECHNICAL RULES

4.1 Design Environment and Technology

In the digital processor design category, Verilog-2005 or System Verilog hardware design language (HDL) must be used for the hardware to be developed. The rules given in section APP-3 must be followed while writing the HDL code. The design must be updated at least once in 20-days periods to a private repository opened on GitHub and shared with the competition evaluation board. Design contents to be uploaded to Github will be determined by DDK. In the chip design phase, either OpenLane flow or commercial EDA tool (e.g. Synopsys, Cadence or Siemens) flow will be used. IC flow evaluations will be done at TT (Typical-Typical) – 25 C – 1.80V corner. Minimum operating frequency should be 100 MHz.

4.2 Design Requirements

4.2.1 Digital Processor Design Category

In the digital processor design category of the competition, a customized processor with the RV32IMAFB_Zicsr Instruction Set Architecture is expected to be designed. During the processor design phase, the constraints specified in Design Environment and Technology section must be complied with. Participants can reuse circuits

available within the open-source design community (appropriate references should be provided).

The features that the processor in question should have are listed below.

- Support for the instructions given in APP-1
- Supporting UART peripheral detailed in APP-2.
- Total 4 KB level 1 cache (L1 Cache) for holding the instructions and the data.
- 128-bit off-chip interface in the processor design for data that is not in level 1 cache (Signals in the interface and the specifications of the transfers will be determined in the top module (wrapper) to be shared by the competition committee.)
- This interface should be used for data from the end address of the peripherals to the end address of the main memory, and data outside the main memory region should be forwarded to the processor without storing into caches
- Only the main memory area, which starts from the address 0x80000000, should be cachable.
- The program counter value at boot time should point to the base address of the main memory

4.3 Design Outputs

Digital processor design category: The expected design outputs are listed below. All these outputs can be generated via standard OpenLane flow. The files given below must be reproducible through the design in the repository created by the competitors on GitHub.

- GDSII (.gds): The industry standard layout file format used for submitting the design to manufacturing. DRC (Design Rule Check) will be performed by reading GDSII (Graphic Design System) via EDA tool. LVS (Layout versus Schematic) will also be done using this file format.
- Gate level netlist after layout (.v): The gate level netlist shows the standard cells and their connections in the design obtained after the chip flow. By using the gate level during the simulation, it will be verified that the design gives the correct outputs against the given inputs even after the chip flow.
- SDF (Standard Delay Format): SDF refers to the delay information of standard cells and the connections between them.

- DEF (.def): The DEF (Design Exchange Format) file contains the location of standard cells in addition to the port level netlist. By reading over the relevant EDA tool , general examinations about the design will be made and field information will be obtained.
- Reports: The design needs to be clean in terms of timing (setup, hold), DRV (maximum capacitance, max slew), and physical verification (DRC, LVS). For this purpose, test/analysis reports should be given.

Gate level simulation will not be counted among the minimum performance criteria, but it will have a point equivalent in the evaluation. For this reason, the relevant file must be submitted.

5 AWARD

As a result of the evaluation, the teams that pass the report stages and reach the finals in their own category, meet the award criteria and are ranked in the final evaluation, will be awarded a monetary award as indicated in Table 2, Table 3 and Table 4. **Categories will be evaluated and awarded separately**. First, second and third place prizes will be divided equally according to the total number of team members (all members registered in the system) and will be deposited into the bank account specified by each person. The relevant category organizer will determine whether the teams that cannot meet the minimum success criteria for the award ranking in the competition categories can receive an honorable mention award or how much they can receive.

The prizes in the table below show the total amount that will be awarded to the teams that are eligible to receive the prize, no individual prizes will be awarded. The first, second and third prizes will be divided equally according to the total number of team members (excluding the consultant) and will be deposited into the bank account specified by each person. Team consultants who are entitled to receive an award cannot benefit from the first, second and third prize amounts below, the awards to be given to the consultants are also stated in the table below.

Table 2: Digital Processor Category Awards

RANK	AWARD AMOUNT	CONSULTANT
First	150.000,00 ₺	9.000,00 ₺
Second	120.000,00 ₺	7.500,00 ₺
Third	100.000,00 ₺	6.000,00 ₺

Table 3: Analog Category Awards

RANK	AWARD AMOUNT	CONSULTANT
First	150.000,00 ₺	9.000,00 ₺
Second	120.000,00 ₺	7.500,00 ₺
Third	100.000,00 ₺	6.000,00 ₺

Table 4: Microcontroller Category Awards

RANK	AWARD AMOUNT	CONSULTANT
First	150.000,00 ₺	9.000,00 ₺
Second	120.000,00 ₺	7.500,00 ₺
Third	100.000,00 ₺	6.000,00 ₺

5.1 Minimum Success Criteria for Award Ranking in Digital Processor Design Category

The minimum success criteria that competitors must meet in order to enter the award ranking are as follows:

- The processor must support all 32-bit instructions given in this document.
- The processor must meet the requirements given in Digital Processor Design Category section that is below the Design Requirements header.
- The checks specified in the Design Outputs section, should be performed to show that there are no violations.
- The processor must successfully run test software prepared in line with the requirements.

In addition, the competitor must have passed to the presentation stage and must be in the festival area on the day of the event.

6 GENERAL RULES

Click here to access the General Rules booklet which is valid for the competition.

7 CODES OF CONDUCT

Click here to access the Code of Ethics booklet that is valid for the competition.

Responsibility Statement

- T3 Foundation and TEKNOFEST are not responsible for any product delivered by the competitors or any injury or damage caused by the competitor. T3 Foundation and organization officials are not responsible for the damages caused by the competitors to third parties. T3 Foundation and TEKNOFEST are not responsible for ensuring that the teams prepare and implement their systems within the framework of the laws of the Republic of Turkiye.

Turkiye Technology Team Foundation reserves the right to make any changes in this specification.

8 APPENDIX

APP-1 Instructions to Be Supported

Table 5 : RV32I Basic Instruction Set

imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[12|10:5]	rs2	rs1	110	imm[4:1|11]	1100011	BLTU
imm[12|10:5]	rs2	rs1	111	imm[4:1|11]	1100011	BGEU
imm[11:0]		rs1	000	rd	0000011	LB
imm[11:0]		rs1	001	rd	0000011	LH
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:0]		rs1	100	rd	0000011	LBU
imm[11:0]		rs1	101	rd	0000011	LHU
imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	SB
imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	SH
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	010	rd	0010011	SLTI
imm[11:0]		rs1	011	rd	0010011	SLTIU
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0100000	shamt	rs1	101	rd	0010011	SRAI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND

Table 6: RV32M Standard Instruction Set Extensions

0000001	rs2	rs1	000	rd	0110011	MUL
0000001	rs2	rs1	001	rd	0110011	MULH
0000001	rs2	rs1	010	rd	0110011	MULHSU
0000001	rs2	rs1	011	rd	0110011	MULHU
0000001	rs2	rs1	100	rd	0110011	DIV
0000001	rs2	rs1	101	rd	0110011	DIVU
0000001	rs2	rs1	110	rd	0110011	REM
0000001	rs2	rs1	111	rd	0110011	REMU

Table 7: RV32A Standard Instruction Set Extensions

00010	aq	rl	00000	rs1	010	rd	0101111	LR.W
00011	aq	rl	rs2	rs1	010	rd	0101111	SC.W
00001	aq	rl	rs2	rs1	010	rd	0101111	AMOSWAP.W
00000	aq	rl	rs2	rs1	010	rd	0101111	AMOADD.W
00100	aq	rl	rs2	rs1	010	rd	0101111	AMOXOR.W
01100	aq	rl	rs2	rs1	010	rd	0101111	AMOAND.W
01000	aq	rl	rs2	rs1	010	rd	0101111	AMOOR.W
10000	aq	rl	rs2	rs1	010	rd	0101111	AMOMIN.W
10100	aq	rl	rs2	rs1	010	rd	0101111	AMOMAX.W
11000	aq	rl	rs2	rs1	010	rd	0101111	AMOMINU.W
11100	aq	rl	rs2	rs1	010	rd	0101111	AMOMAXU.W

Table 8: RV32F Standard Instruction Set Extensions

RV32F Standard Extension

imm[11:0]			rs1	010	rd	0000111	FLW
imm[11:5]		rs2	rs1	010	imm[4:0]	0100111	FSW
rs3	00	rs2	rs1	rm	rd	1000011	FMADD.S
rs3	00	rs2	rs1	rm	rd	1000111	FMSUB.S
rs3	00	rs2	rs1	rm	rd	1001011	FNMSUB.S
rs3	00	rs2	rs1	rm	rd	1001111	FNMADD.S
0000000		rs2	rs1	rm	rd	1010011	FADD.S
0000100		rs2	rs1	rm	rd	1010011	FSUB.S
0001000		rs2	rs1	rm	rd	1010011	FMUL.S
0001100		rs2	rs1	rm	rd	1010011	FDIV.S
0101100		00000	rs1	rm	rd	1010011	FSQRT.S
0010000		rs2	rs1	000	rd	1010011	FSGNJ.S
0010000		rs2	rs1	001	rd	1010011	FSGNJN.S
0010000		rs2	rs1	010	rd	1010011	FSGNJX.S
0010100		rs2	rs1	000	rd	1010011	FMIN.S
0010100		rs2	rs1	001	rd	1010011	FMAX.S
1100000		00000	rs1	rm	rd	1010011	FCVT.W.S
1100000		00001	rs1	rm	rd	1010011	FCVT.WU.S
1110000		00000	rs1	000	rd	1010011	FMV.X.W
1010000		rs2	rs1	010	rd	1010011	FEQ.S
1010000		rs2	rs1	001	rd	1010011	FLT.S
1010000		rs2	rs1	000	rd	1010011	FLE.S
1110000		00000	rs1	001	rd	1010011	FCLASS.S
1101000		00000	rs1	rm	rd	1010011	FCVT.S.W
1101000		00001	rs1	rm	rd	1010011	FCVT.S.WU
1111000		00000	rs1	000	rd	1010011	FMV.W.X

Table 9: RV32B Standard Instruction Set Extensions

The designed processor must support the instructions where RV32 is marked in the table.

RV32	RV64	Mnemonic	Instruction	Zba	Zbb	Zbc	Zbs
	✓	add.uw *rd, rs1, rs2*	Add unsigned word	✓			
✓	✓	andn *rd, rs1, rs2*	AND with inverted operand		✓		
✓	✓	clmul *rd, rs1, rs2*	Carry-less multiply (low-part)			✓	
✓	✓	clmulh *rd, rs1, rs2*	Carry-less multiply (high-part)			✓	
✓	✓	clmulr *rd, rs1, rs2*	Carry-less multiply (reversed)			✓	
✓	✓	clz *rd, rs*	Count leading zero bits		✓		
	✓	clzw *rd, rs*	Count leading zero bits in word		✓		
✓	✓	cpop *rd, rs*	Count set bits		✓		
	✓	cpopw *rd, rs*	Count set bits in word		✓		
✓	✓	ctz *rd, rs*	Count trailing zero bits		✓		
	✓	ctzw *rd, rs*	Count trailing zero bits in word		✓		
✓	✓	max *rd, rs1, rs2*	Maximum		✓		
✓	✓	maxu *rd, rs1, rs2*	Unsigned maximum		✓		
✓	✓	min *rd, rs1, rs2*	Minimum		✓		
✓	✓	minu *rd, rs1, rs2*	Unsigned minimum		✓		
✓	✓	orc.b *rd, rs1, rs2*	Bitwise OR-Combine, byte granule		✓		
✓	✓	orn *rd, rs1, rs2*	OR with inverted operand		✓		
✓	✓	rev8 *rd, rs*	Byte-reverse register		✓		
✓	✓	rol *rd, rs1, rs2*	Rotate left (Register)		✓		
	✓	rolw *rd, rs1, rs2*	Rotate Left Word (Register)		✓		
✓	✓	ror *rd, rs1, rs2*	Rotate right (Register)		✓		
✓	✓	rori *rd, rs1, shamt*	Rotate right (Immediate)		✓		
	✓	roriw *rd, rs1, shamt*	Rotate right Word (Immediate)		✓		
	✓	rorw *rd, rs1, rs2*	Rotate right Word (Register)		✓		

RV32	RV64	Mnemonic	Instruction	Zba	Zbb	Zbc	Zbs
✓	✓	bclr *rd, rs1, rs2*	Single-Bit Clear (Register)				✓
✓	✓	bclri *rd, rs1, imm*	Single-Bit Clear (Immediate)				✓
✓	✓	bext *rd, rs1, rs2*	Single-Bit Extract (Register)				✓
✓	✓	bexti *rd, rs1, imm*	Single-Bit Extract (Immediate)				✓
✓	✓	binv *rd, rs1, rs2*	Single-Bit Invert (Register)				✓
✓	✓	binvi *rd, rs1, imm*	Single-Bit Invert (Immediate)				✓
✓	✓	bset *rd, rs1, rs2*	Single-Bit Set (Register)				✓
✓	✓	bseti *rd, rs1, imm*	Single-Bit Set (Immediate)				✓
✓	✓	sext.b *rd, rs*	Sign-extend byte		✓		
✓	✓	sext.h *rd, rs*	Sign-extend halfword		✓		
✓	✓	sh1add *rd, rs1, rs2*	Shift left by 1 and add	✓			
	✓	sh1add.uw *rd, rs1, rs2*	Shift unsigned word left by 1 and add	✓			
✓	✓	sh2add *rd, rs1, rs2*	Shift left by 2 and add	✓			
	✓	sh2add.uw *rd, rs1, rs2*	Shift unsigned word left by 2 and add	✓			
✓	✓	sh3add *rd, rs2, rs2*	Shift left by 3 and add	✓			
	✓	sh3add.uw *rd, rs1, rs2*	Shift unsigned word left by 3 and add	✓			
	✓	slli.uw *rd, rs1, imm*	Shift-left unsigned word (Immediate)	✓			
✓	✓	xnor *rd, rs1, rs2*	Exclusive NOR		✓		
✓	✓	zext.h *rd, rs*	Zero-extend halfword		✓		

APP-2 Peripheral Details

Peripherals must be built with peripheral bus and devices that support the peripheral bus protocol. The peripheral bus protocol must be selected among three options: Wishbone, AXI4-Lite, and TileLink Uncached Lightweight (TL-UL). One peripheral host and three devices must be supported.

It is desired to design a single peripheral, Universal Asynchronous Receiver / Transmitter (UART). A memory map should be created where peripheral controls will be carried out. Memory maps should be designed to require only 32-bit memory accesses. In order for your designs to integrate seamlessly on test systems, all the specifications for peripherals must be supported.

UART (Universal Asynchronous Receiver Transmitter)

In this section, the basic features of the UART device to be designed are specified.

The UART module should be a serial-to-parallel (rx) and parallel-to-serial (tx) full-duplex design, typically intended to communicate with an external device for terminal style communication. Programmable baud rate up to 1 Mbps must be guaranteed.

Features:

- 1 start bit, 8 data bits, no parity bit, 1 stop bit
- Programmable baud rate
- 32 x 8-bit rx buffer
- 32 x 8-bit tx buffer
- rx input port
- tx output port

Memory Map:

The memory map for the UART peripheral control registers is shown in Table 10.

Table 10: UART Memory Map

Address	Name	Description
0x20000000	uart_ctrl	Control Register
0x20000004	uart_status	Status Register
0x20000008	uart_rdata	Data Read Register
0x2000000c	uart_wdata	Data Write Register

UART Control Register

Controls the operation of the tx and rx channels.

Table 11: UART Control Register

uart_ctrl reset default: 0x0, mask: 0xFFFF0003			
Bits	**Type**	**Name**	**Description**
0	read / write	tx_en	tx enable
1	read / write	rx_en	rx enable
15:2			Unused space
31:16	read / write	baud_div	Baud rate control

tx_en should check if tx channel is enabled or not. When enabled, transmission should be performed if there is data in the tx buffer.
rx_en should check if the rx channel is enabled. When enabled, the incoming data should be kept in the rx buffer until the read operation takes place.

For both tx and rx channels, the baud rate must be specified with **baud_div**. The relationship between the input clock frequency and the baud should be established with the following equation.

$$f_{baud} = \frac{f_{clk}}{baud\_div + 1}$$

Table 12 gives examples of division values to obtain a particular baud rate at a particular clock frequency.

Table 12: Example of baud rate calculation values

Clock (MHz)	Target Baud (Hz)	baud_div	Actual Baud (Hz)
500	31250	16000	31250
500	115200	4340	115207
500	250000	2000	250000
500	1843200	271	1845018

UART Status Register

This register gives information about the rx and tx buffer states.

Table 13: UART Status Register

uart_status reset default: 0xA, mask: 0xF			
Bits	**Type**	**Name**	**Description**
0	read only	tx_full	tx buffer is full
1	read only	tx_empty	rx buffer is empty
2	read only	rx_full	rx buffer is full
3	read only	rx_empty	rx buffer is empty

UART Data Read Register

The data received serially from the rx channel is read in parallel with this register. After rx_en is enabled, data must be stored in the rx buffer until read from this register.

Table 14: UART Data Read Register

uart_rdata reset default: 0x0, mask: 0xFF			
Bits	**Type**	**Name**	**Description**
7:0	read only	rdata	Data read

UART Data Write Register

The data to be transmitted over the tx channel is written to this register and serial transmission is performed. Data write requests should be stored in the tx buffer when tx_en is not enabled, and data should be transmitted when the channel is active.

Table 15: UART Write Data Register

uart_wdata reset default: 0x0, mask: 0xFF			
Bits	**Type**	**Name**	**Description**
7:0	write only	wdata	Data write

APP-3 Verilog RTL Writing Rules

In order to make the code easier to read and understand in the competition, the following guidelines should be followed.

- Each .v file should consist of a single module and the name of the file should be the same as the name of the module.
- Expressions such as always, if, else should start with begin and end with end.
- Indents should consist of 3 spaces.
- Clocks should start with clk. Additional attachments can be added upon request. eg; clk_memory.
- Resets should start with rst. Additional attachments can be added upon request. eg; rst_memory.
- The inside section in begin / end, module / endmodule, case / endcase should be indented from the previous section. Sample code is given below.

```
always @(posedge clk or negedge rst) begin
   if (!rsti) begin
      valid1 <= 1'b0;
   end else begin
      valid1 <= valid0;
   end
end
```

- Module inputs, outputs, and inouts should end with "_i", "_o", and "_io", respectively.
- Module expressions should be in Verilog-2001 format. After the "module" statement, the module name, input/output, signal should be in parentheses, and then the logic part of the module should be written. Sample code is given below.

```
module modul_ismi #(
   parameter int Size = 8
) (
   input                    clk_i,
   input                    rst_i,
   input   [Size-1:0]     rdata0_i,
   input   [Size-1:0]     rdata1_i,
   output [Size-1:0]   wdata0_o,
   output [Size-1:0]   wdata1_o
);
   …
endmodule
```

- When calling the module, it should not be called in a single line, but the parameters and the signal names should be written clearly one under the other. The unconnected input signal should be connected to ground and the unconnected output signal should be left blank. Sample code is given below.

```
…
modul_ismi #(
  . Size(16)
) örnek_isim (
  .clk_i (clk),
  .rst_i(1'b0), //unconnected empty input signal must be connected to
the soil
  .rdata0_i (rdata) ,
  .rdata1_i (16'b0),//unconnected empty input signal must be connected
to the soil
  .wdata0_o (wdata),
  .wdata1_o () //unconnected idle output signal must be left blank
```

- When assigning two signals/ports to each other, the signal/port length should be the same. Automatic assignment should not be made.

```
wire [3:0] ornek_sinyal;
wire [6:0] ornek_sinyal2;
ornek_sinyal = 4'd4; = 4 gives the same, but the length is not the same.
ornek_sinyal2 = {3'b0,ornek_sinyal}; = ornek_sinyal gives the same, but the length is not the same.
```

- Combination blocks should only use blocking assignments (=).
- Sequential logic blocks should only use non-blocking assignments (<=).
- Signal names should be meaningful. Long and meaningful names should be preferred over short names.
- Comment lines should be added where necessary to clarify the code.