CHIP DESIGN COMPETITION SPECIFICATIONS 2024

CHANGE TRACKING TABLE

Version	Date	Change Place	Description
1.0	07.12.2024	2024 First Publication	
1.1	20.02.2024		Update of the application deadline under the Competition Schedule heading
1.2	26.02.2024		Add “APP-4 Microcontroller Peripheral Registers”
1.3	27.03.2024		Add USB FS Device and JTAG peripheral information at “APP-4”
1.4	03.04.2024		Update under the Competition Schedule heading
1.5	15.04.2024		Update under the Competition Schedule heading

TABLE OF CONTENTS

FIGURE AND TABLE LIST

ABBREVIATIONS

ADC: Analog-to-Digital Converter
ALU: Arithmetic Logic Unit
BILGEM: Informatics and Information Security Research Center
BP: Band Pass
CNN: Convolutional Neural Network
DC: Direct Current
DCT: Direct Cosine Transform
DDK: Advisory and Evaluation Board
DEF: Design Exchange Format
DRC: Design Rule Check
DRV: Design Rule Violation
DSP: Digital Signal Processing
DDR: Detailed Design Report
EDA: Electronic Design Automation
FPGA: Field Programmable Gate Array
GDS: Graphic Data System
GPS: Global Positioning System
HDL: Hardware Description Language
IIP3: Input Third Order Intercept Point
IPC: Instruction per Cycle
IRN: Input Referred Noise
ISA: Instruction Set Architecture
IP: Intellectual Property
KYS: Enterprise Management System
LDO: Low Drop-out
LTE: Long Term Evolution
LVS: Layout versus Schematic
MCU: Microcontroller Unit
MPW: Multi Project Wafer
PDK: Process Design Kit
PDR: Preliminary Design Report
PSRR: Power Supply Rejection Ratio
QVGA: Quarter Video Graphics Array
RISC: Reduced Instruction Set Architecture
RF: Radio Frequency
RTL: Register Transfer Level
SDF: Standard Delay Format
SoC: System-on-Chip
SV: SystemVerilog
T3 Foundation: Turkiye Technology Team Foundation
TT: Typical- Typical
TÜBİTAK: Scientific and Technological Research Council of Turkiye
UART: Universal Asynchronous Receiver Transmitter

USB: Universal Serial Bus
UVM: Universal Verification Methodology
VCO: Voltage Controlled Oscillator
WIPO: World Intellectual Property Organization

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	29.02.2024
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	22.04.2024
Submission Deadline for Detail Design Report (DDR)	30.06.2024 17.00
Announcement of Finalist Teams	15.07.2024
QA Session	22.07.2024
Deadline for Finalizing the Design	Not yet determined
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 29, 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.

In the analog design category detailed design report, it is expected that the schematic level design will be completed and reported with the results.

Microcontroller Unit (MCU) category detailed design report should include topics such as how the requirements presented in the specification are realized, block diagrams, simulation and synthesis results. The processor core in the MCU and the "Spike" RISC-V instruction set simulation tool must be verified by comparing the results of the programs provided by DDK. In addition, it is expected that the test results and coverage report of the UART peripheral unit that is requested to be verified with SV/UVM will be detailed. While SV/UVM verification must be shown to be working correctly for other peripherals, SV/UVM verification is not required, only SV/UVM is required for UART.

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***.***

3.3.2 Analog Design Category

Scoring will be made out of 100 points and reports, presentations, and demos will constitute the entire score. Reports are 40% of the total score and the remaining 60% will be presentation and performance criteria. Evaluation will be made through post-layout simulations in the analog design category.

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 Score = (0.15 * PDR Points) + (0.25 * DDR Points) + (0.40 * Presentation Points) + (0.20 * Performance Criteria Points)

Performance Criteria Points will be determined by considering the performances, area and power consumptions of the bandgap reference voltage circuit, LDO voltage regulator circuit and voltage controlled oscillator (VCO) circuit.

3.3.3 Microcontroller Design Category

The module, which will be designed within the scope of the microcontroller design category, will consist of two separate parts: FPGA and physical design realization. The competitors will first create a design on an FPGA development board that will fulfill the tasks given in the Microcontroller section of the Microcontroller Design Category under the title of Design Requirements and run this design in real-time on the FPGA card. In the next stage, they will create GDSII outputs in the form to be sent to production with the IC design software that will be provided for physical design. Detailed information about the IC physical design will be clarified in the future.

Technical information about what needs to be done for the FPGA and Physical Design stages is defined in the Microcontroller Design Category heading under the TECHNICAL RULES heading.

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 * Verification Points)

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.

In the analog design category, commercial-grade EDA tools (vendors like Synopsys, Cadence, Siemens) and its appropriate PDK will be used. Version information for the PDK and the tools will be shared in a later announcement. The designs will be updated to a special repository created on GitHub before the delivery of the detail design report and final presentation and will be shared with the competition evaluation board.

In the microcontroller design category, for the hardware to be developed Verilog or SystemVerilog hardware design language (HDL) must be used. The design has been shared with the competition evaluation board and must be updated at least every 20 days in a special repository opened on GitHub. Design contents to be uploaded to Github will be determined by DDK. During the chip design phase, an open source (OpenLane) or commercial EDA tool (such as Synopsys, Cadence, Siemens) and PDK suitable for these tools will be used. Details will be shared by DDK at a later date. The UVM library and SystemVerilog HDL should be used for UART peripheral verification. When verifying the processor core defined in the specification that will be used in the SoC, the test programs should be shown to be working correctly by comparing them with the Spike ISS results.

GitHub repositories of the participating teams should be shared with the "TUTEL-TUBITAK" account for the digital processor design and analog design category, and with the "yongatek-teknofest" account for the microcontroller design category. Contestants are required to share their GitHub repositories with the relevant accounts from the date to be determined later. Otherwise, they will lose points as stated in the scoring section. The date from which the repository should begin to be shared will be announced later.

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

4.2.2 Analog Design Category

Today, the trend towards portable devices requires that many electronic systems be designed to operate with batteries. Power management circuits designed in this direction are implemented in integrated circuits instead of discrete elements in order to meet increased performance, low power consumption and space demand.

Figure 1 shows a representative time-voltage graph of a battery. Some batteries discharge almost linearly while powering the circuit due to their characteristic. When the circuits are fed directly with the battery, the performance of the circuit also changes over time as a result of the time-varying supply voltage and deviates from the optimum performance. In order to avoid this problem, power converters are used to convert the battery voltage to a constant voltage and provide it to the circuit.

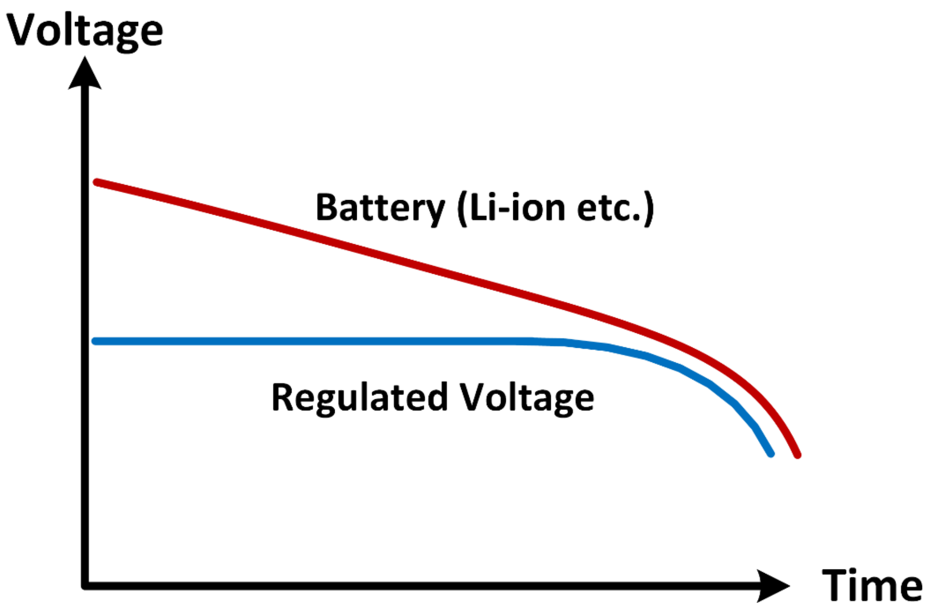


Figure 1: Battery Discharge Graph

LDO (Low-Dropout) voltage regulator circuit is one of the commonly used power converter types to obtain a constant and noiseless voltage in battery powered systems. It is preferred because of its simpler structure and more affordable cost compared to other converter types. It is frequently used in battery powered systems such as automotive, mobile phones and laptop computers. The operation of this circuit is that it gives the variable or noise voltage at the input as a constant voltage at the output.

Designers are requested to design an LDO (Low-Dropout) voltage regulator and a voltage-controlled oscillator (VCO) to be powered by this regulator. VCO will have technical requirements that can be used in L1 and L2 GPS systems. It is also expected that the bandgap reference circuit will be designed to generate the reference voltage of the LDO voltage regulator. In Figure 2, the blocks to be designed within the scope of the competition are visualized.

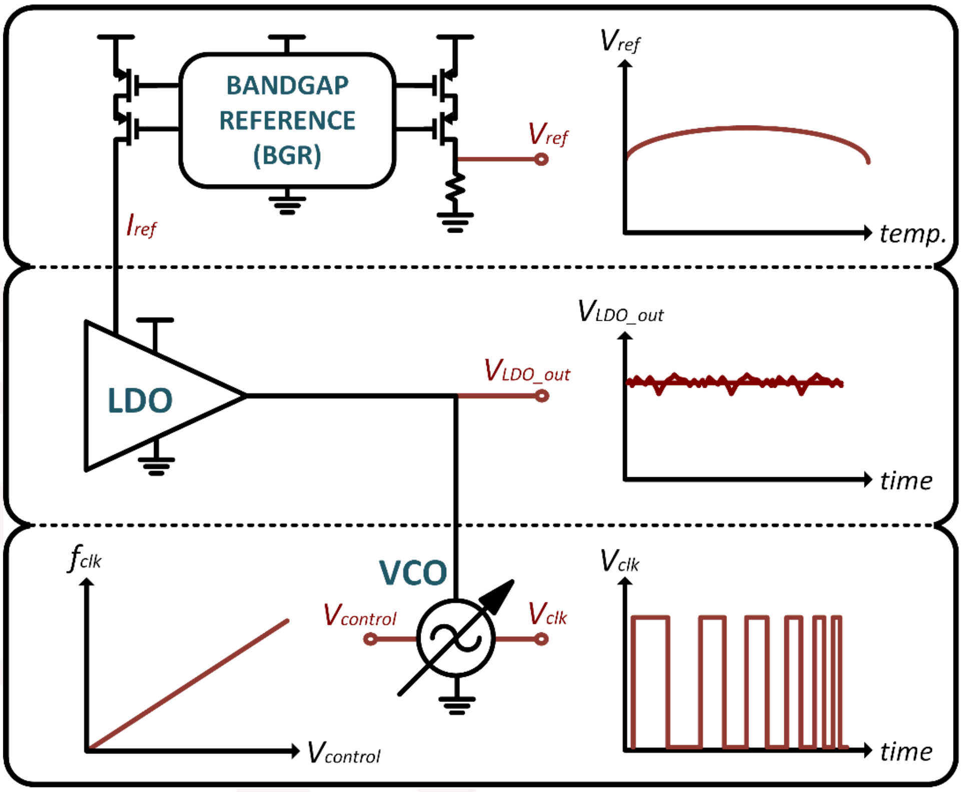


Figure 2: Block Diagram of Bandgap Reference, LDO Voltage Regulator and Voltage Controlled Oscillator Circuits

Block level and general design requirements are given below in detail.

General Requirements:

- The design environment and technology specified in Section 3.1 must be complied with.
- The band-gap reference and LDO voltage regulator circuits should be powered by only one ideal voltage source of 1.8 V.
- All circuit elements should be used from the process library, ideal elements should not be used.
- Simulations should be made at the TT (Typical-Typical) process corner and at 27°C. The 27°C temperature is the nominal temperature at which the entire design will operate. Only simulations of the bandgap reference voltage circuit should cover the temperature range of -40°C to 125°C.
- Verify that the following requirements are met by post-layout simulations.
- The schematic and post-layout simulation results should be presented comparatively.
- Except for start-up circuits, all MOS transistors must operate in the saturation region. (Applicable to bandgap reference and LDO voltage regulator circuits)
- Stability analysis should be performed when necessary.

LDO regulator requirements:

- Input voltage should be 1.8V, output voltage should be 1.2V.

- The output voltage should show a maximum variation of 300 mV against current pulses with a pulse-period ratio of 50% from 0 mA to 50 mA. (The current pulse frequency must be 500 kHz and the current pulse source must be connected so that current flow from the regulator output to ground.)
- Loop gain and loop phase margin values should be minimum 45 dB and 50°.
- The above requirements should be tested by connecting an ideal 500 pF capacitor to the output. This capacitor is for the test setup and should not be included in the circuit.
- If there is no load at its output, the power consumption of the LDO voltage regulator should be a maximum of 1.8 mW.
- Line regulation should be checked by making the input voltage 2.5V.
- Load regulation should be checked by drawing a constant 10mA and 20mA current from the output.

Voltage-controlled oscillator (VCO) requirements:

- The VCO must be powered by the LDO voltage regulator and the supply voltage must be 1.2V.
- A decoupling capacitor can be connected to the VCO supply input. (It should be in VCO layout if it's connected.)
- The load to be driven at the VCO output must be at least 0.4pF. This capacitance should only be connected in the test schematic. It shouldn't be in the VCO block. Therefore, it doesn’t need to be in the VCO layout.
- The VCO must be able to drive this load in a square wave form between 0 and 1.2V (rail-to-rail)
- VCO operating frequency should be between 1200 – 1600 MHz.
- Kvco value shouldn’t be greater than 800 MHz/V.
- Phase Noise value must be better than 75 dBc for 1 MHz offset.
- The VCO control voltage range should be in the range of 0 - 1.2 V.
- The duty cycle value for the square wave obtained while driving the VCO load should be between 40%-60%.
- The sum of "rise time" + "fall time" for the square wave obtained while driving a load of 0.4pF at 1600MHz frequency should be less than 0.2ns. (For the rise time, should be looked at the time when the amplitude of the square wave increases from 10% to 90%. For the fall time, should be looked at the time when the amplitude of the square wave decreases from 90% to 10%.)

Bandgap reference voltage circuit requirements:

- A 500 mV reference voltage should be generated as shown in Figure 2 for the purpose of validating the bandgap reference circuit.
- The reference voltage should vary by a maximum of 20 ppm/°C, covering the temperature range of -40°C to 125°C.
- For 1.8±10% supply voltage, the 500mV reference voltage should not change and the transistors should protect their nominal operating regions.
- A start-up circuit should be utilized to avoid circuit startup problems.
- The PSRR value should be below 0 dB in the 1 Hz-10 GHz range.

4.2.3 Microcontroller Design Category

Two solutions will be realized under the microcontroller category: The first will work on FPGA (FPGA prototyping), the second will be to create production files with the physical design tool (IC Design).

The microcontroller to be designed by the competitors will use the CV32E40P RISC-V core IP with a single-core, 32-bit 4-stage pipeline, which is maintained by the "OpenHW Group" and available as open source in the github environment. The github repo link of the relevant processor core is given below:

https://github.com/openhwgroup/cv32e40p

The documentation link for the CV32E40P processor core is given below:

https://docs.openhwgroup.org/projects/cv32e40p-user-manual/en/latest/index.html

Which release and commit version and which configuration of the IP will be used within the scope of the competition, that is, which commands it will support or not, will be clarified by DDK at a later date.
The microcontroller will have the following peripherals:

1x UART

1x I2C Master

1x QSPI Master

1x Timer

1x GPIO (32 pin I/O)

1x USB Full-Speed Device (12 Mbps)

1x JTAG (Optional)

Register definitions and other details regarding peripheral units are defined under the heading "APP-4 Microcontroller Peripheral Registers".

Competitors can design peripheral designs themselves, or they can use open-source projects by referencing them.

The following open-source repos are provided by DDK as examples for USB Device peripheral design:

https://github.com/WangXuan95/FPGA-USB-Device

https://github.com/ulixxe/usb_cdc

In USB Device design, competitors are expected to realize USB Communication Device Class (USB-CDC) functionality. Other device class functions (audio, HID, mass storage, video, etc.) will be optional and will provide additional points.

Validation of the CV32E40P RISC-V processor core is expected to be done with an instruction set emulator (ISP) (e.g. Spike ISS). Spike ISS github repo link:

https://github.com/riscv-software-src/riscv-isa-sim

As part of microcontroller verification, the UART peripheral will be expected to be done using SystemVerilog HDL and Universal Verification Methodology (UVM). In the verification made with UVM, regression and coverage results must be reported. The environment in which UVM will be used (Synopsys, Cadence, Mentor) will be clarified by DDK at a later date.

It is not necessary to use UVM/SV for other peripherals other than UART. Verification can also be performed with other methods and methodologies. However, all peripherals and the MCU top module must be verified.

MCU will contain 8 kB instruction and 8 kB data memory.

Addressing of peripherals and memories will be defined and designed by the competitors.

When the system is powered on, it will boot from a non-volatile memory (QSPI Flash Memory) via the QSPI master interface. The bootloader code will be located in a small ROM (e.g. 512 Bytes or 1 kB) as a piece of code that cannot be changed. When the system is powered on, the Bootloader code will be run first. This piece of code will read the program to be run from Flash memory via the QSPI Master interface and write it to the instruction memory. Afterwards, it will transfer the execution to the instruction memory.

Optionally, a debug module connected to the JTAG interface will be designed. The debug module will be connected to the "debug" port of the CV32E40P processor core and the program running on the MUC can be debugged through this debug module.

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.

The expected design outputs for **analog design category** are as follows:

- GDSII (.gds) (GDS: Graphic Design System): The industry standard layout file format used for submitting the design to production. It will be used for layout evaluation and physical verifications (DRC/LVS).
- Design and test environment schematics: These are the files created by EDA tool Schematic Editor that contain the transistor-level designs and the test environment related to these designs. All schematic files used in the project should be included in the printouts to validate my designs.
- Schematic and post-layout transistor level netlist (.spice/.scs/.l): The transistor level netlist is the file showing the PDK elements in the design and their connections with each other. In the post-deployment netlist, parasitic elements are also included in this file. This file will be used for pre- and post-layout performance verifications.
- Layout output files: EDA tool Layout Editor output file. It is requested for the control of the layout.
- Physical Verification Reports: It consists of reports containing the results of physical verification tests (DRC/LVS).

Teams are required to upload their work to a remote repository or server location determined by DDK by a date to be specified later. Announcement about the upload format will be shared with the teams.

For the **microcontroller design category**, different outputs are expected for FPGA and physical design flows. The outputs are expected to be in the GitHub account, which the teams will share with the microcontroller design category regulatory company.

Different outputs are expected for the FPGA and physical design flow for the **microcontroller design category**. The outputs are expected to be in the GitHub account that the teams will share with the DDK.

FPGA Flow

- All RTL Verilog/SV design codes
- All testbench Verilog/SV codes
- Successful (error free) synthesis result report

- Successful (error free) stating timing analysis report
- Successful (error free) implementation (Place & Route) result report
- Bitstream that can be loaded into the FPGA

IC Physical Design Flow

- All RTL Verilog/SV design codes
- All testbench Verilog/SV codes
- Successful completion of physical design flow
- Production ready LEF, DEF, GDSII file outputs of the design
- Successful completion of physical design signoff tests (LVS, DRC, ERC, etc.)

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.

5.2 Minimum Success Criteria for Award Ranking in Analog Design Category

Competitors must meet following specifications at post-layout simulations in order to enter the award ranking:

- Functional operation of the all circuits must be correct.
- LDO loop gain and loop phase margin
- VCO operating frequency and pulse-period ratio
- The temperature coefficient and PSRR value of the band gap reference circuit

Also, the competitors must have passed the presentation stage and be present in the festival area on the day of the event.

5.3 Minimum Success Criteria for Award Ranking in Microcontroller Design Category

The minimum success criteria required for the contestants to be eligible for an award are given below.

- Demonstration that the microcontroller successfully carries out the test scenarios given by the DDK on the FPGA development board.
- Verifying the UART peripheral in the microcontroller with UVM/SV and creating regression and coverage test reports in this context.
- Preparing GDSII outputs ready for production by successfully completing the physical design flow.

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

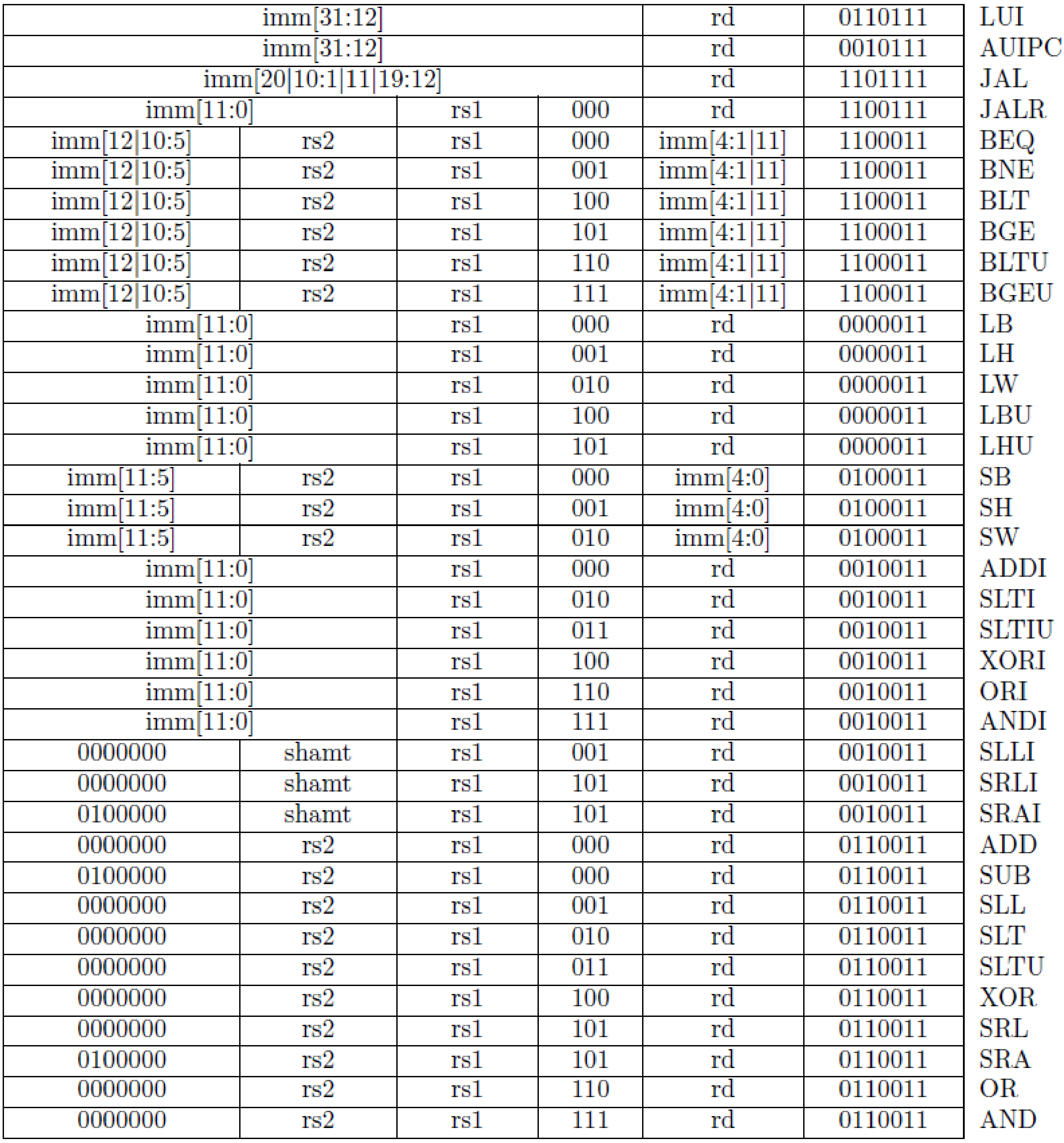
- T3 Foundation and TEKNOFEST are not responsible for any product delivered by the competitors or any injury or damage caused by the competitor. T3 Foundation and organization officials are not responsible for the damages caused by the competitors to third parties. T3 Foundation and TEKNOFEST are not 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

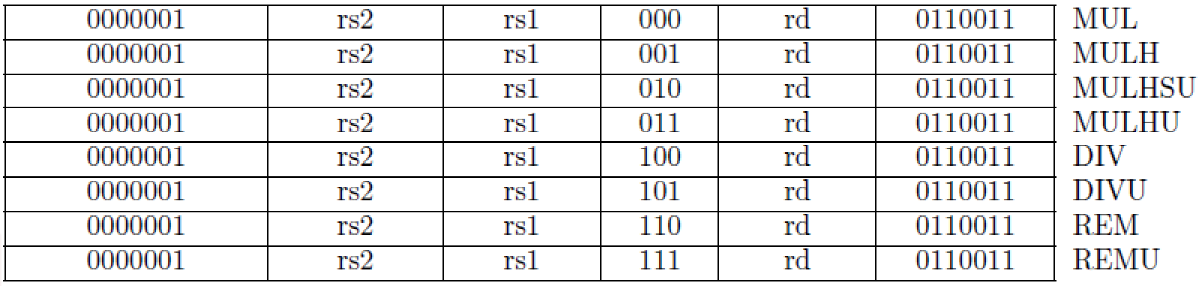
APP-1 Instructions to Be Supported

Table 5 : RV32I Basic Instruction Set



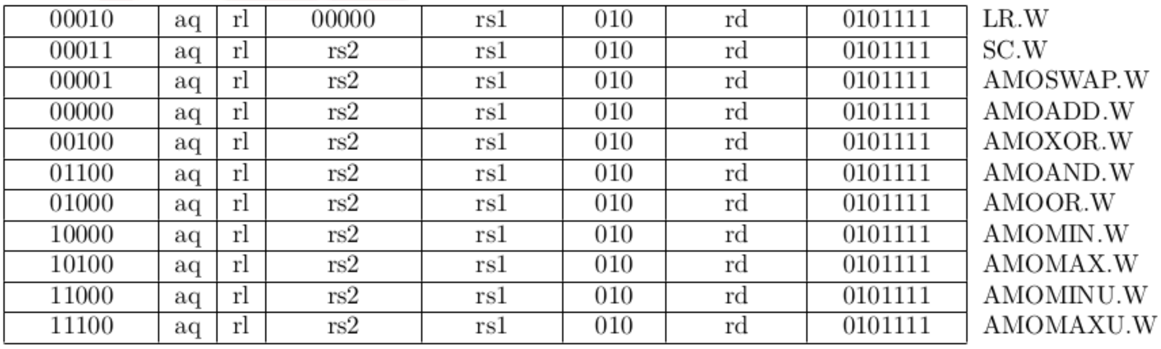
imm[31:12]				rd	0110111	LUI
imm[31:12]				rd	0010111	AUIPC
imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	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



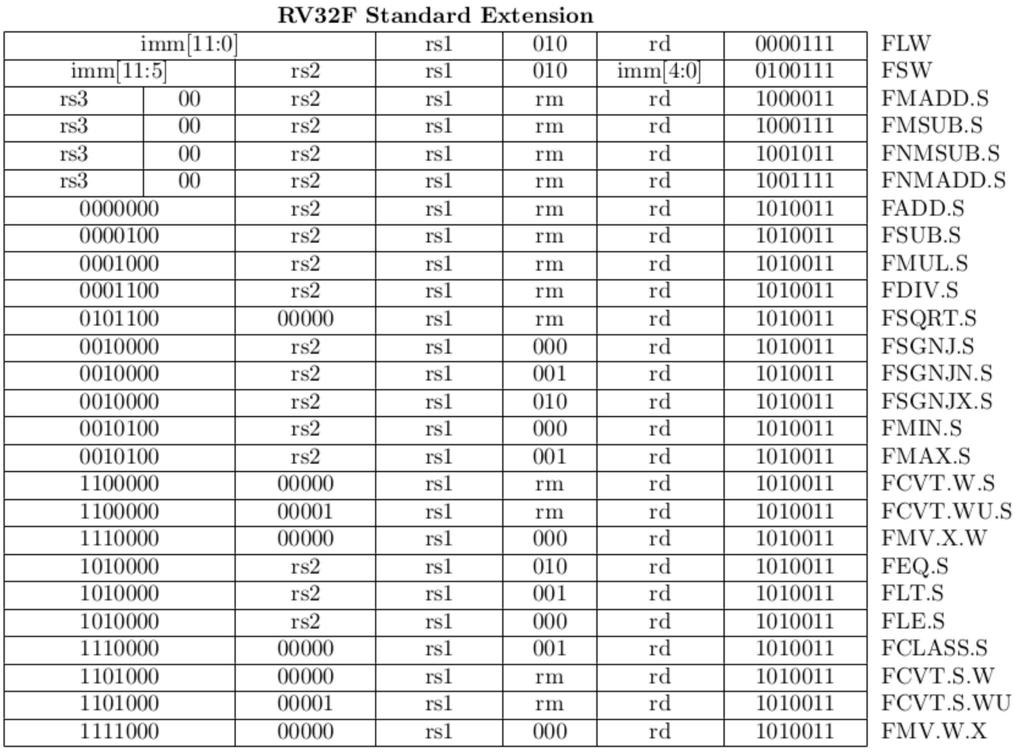
0000001	rs2	rs1	000	rd	0110011	MUL
0000001	rs2	rs1	001	rd	0110011	MULH
0000001	rs2	rs1	010	rd	0110011	MULHSU
0000001	rs2	rs1	011	rd	0110011	MULHU
0000001	rs2	rs1	100	rd	0110011	DIV
0000001	rs2	rs1	101	rd	0110011	DIVU
0000001	rs2	rs1	110	rd	0110011	REM
0000001	rs2	rs1	111	rd	0110011	REMU

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

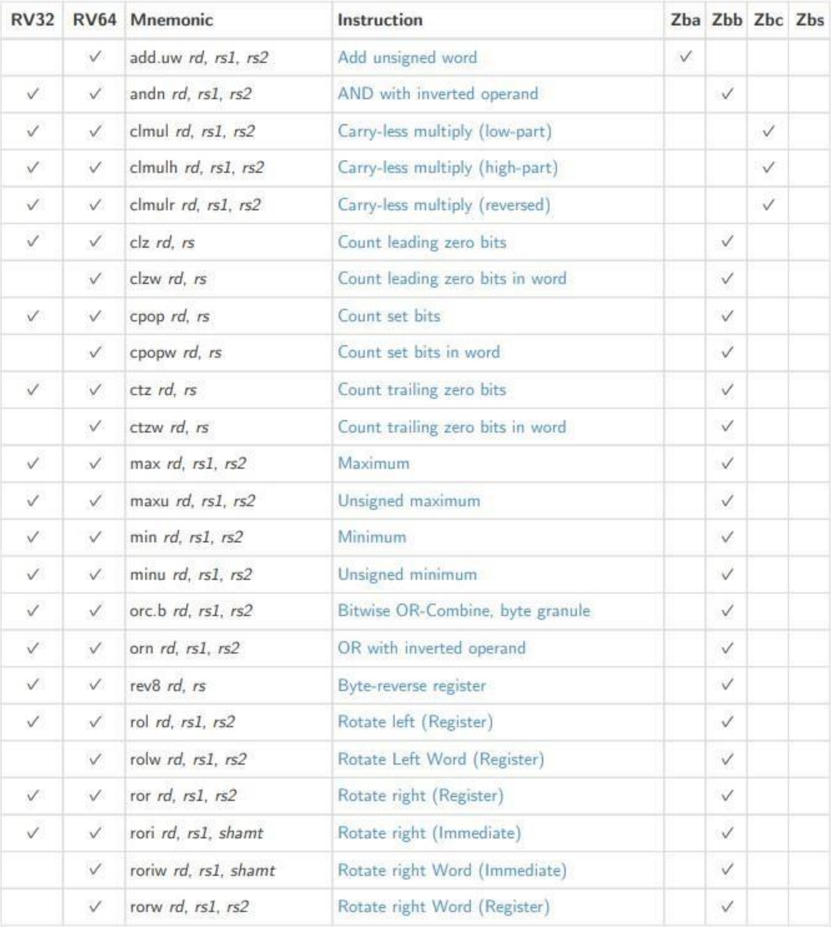


RV32F Standard Extension

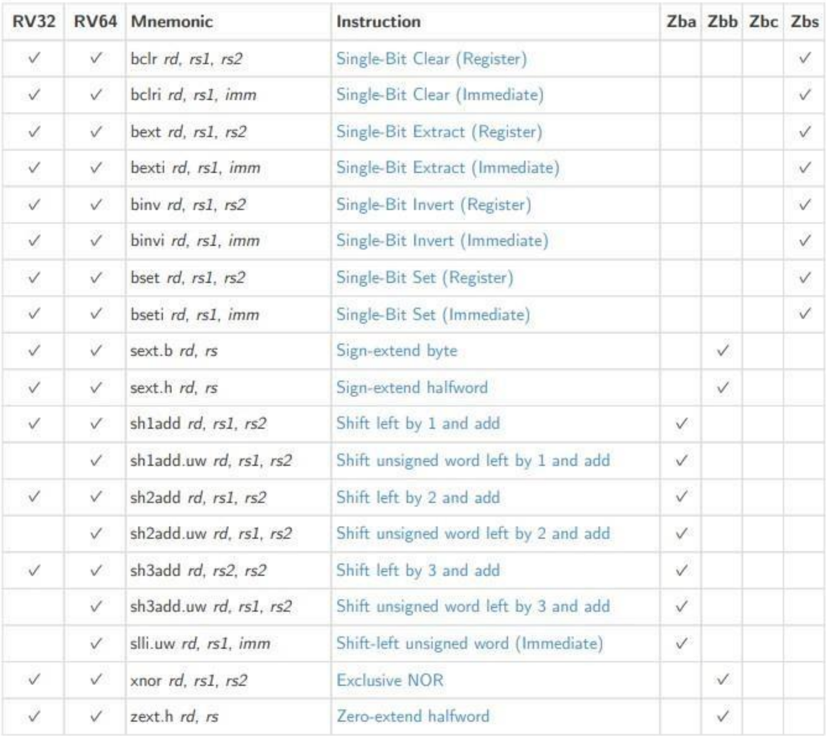
imm[11:0]			rs1	010	rd	0000111	FLW
imm[11:5]		rs2	rs1	010	imm[4:0]	0100111	FSW
rs3	00	rs2	rs1	rm	rd	1000011	FMADD.S
rs3	00	rs2	rs1	rm	rd	1000111	FMSUB.S
rs3	00	rs2	rs1	rm	rd	1001011	FNMSUB.S
rs3	00	rs2	rs1	rm	rd	1001111	FNMADD.S
0000000		rs2	rs1	rm	rd	1010011	FADD.S
0000100		rs2	rs1	rm	rd	1010011	FSUB.S
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 busprotocol. 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 besupported.

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-duplexdesign, 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 followingguidelines should be followed.

- Each .v file should consist of a single module and the name of the file should be thesame 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 beindented 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 thesame. 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 preferredover short names.
- Comment lines should be added where necessary to clarify the code.

APP-4 Microcontroller Peripheral Registers

Peripherals and register definitions defined for the microcontroller are given in APP-4. The bus architecture and protocol for the connections of peripheral units are left to the choice of the competitors. Any standard interface (e.g. AXI4, APB, Wishbone, etc.) can be used. The definition of the base addresses of the peripheral units is also left to the competitors.

GPIO

It is the peripheral unit responsible for reading and writing the values of a total of 32 input and output pins. 16 pins are fixed as input and 16 pins as output.

Offset	Name	Description	R/W
0x00	GPIO_IDR	Input data register. It holds the value of the 16-bit input signal in GPIO_IDR[15:0] bits. The value '0' is always present in the GPIO_IDR[31:16] bits.	Read Only
0x04	GPIO_ODR	Output data register. It transmits the 16-bit value written to the GPIO_ODR[15:0] bits to the output bits. The value written to bits GPIO_ODR[31:16] has no effect.	R/W

TIMER

It is a peripheral unit related to 32-bit counter value. Its value increases and decreases according to the system clock frequency.

Offset	Name	Description	R/W
0x00	TIM_PRE	Prescaler register. The value of the counter register will be changed by dividing the clock frequency by the value written to the TIM_PRE register. For example, when TIM_PRE is '0', the counter (TIM_CNT) will increase/decrease at the system clock rate. In other words, it will change 1 in 1 period of the system clock. When TIM_PRE is '1', the counter will change 1 every 2 periods of the system clock. When TIM_PRE is “0xFFFFFFFF” the counter (TIM_CNT) will change 1 per second.	R/W
0x04	TIM_ARE	Auto-relaod register. The counter register (TIM_CNT) will take the value 0 when it reaches the TIM_AR value and will continue its normal operation in the next period. For example, if the TIM_AR register is 0x36, it will take the value 0 in the period after TIM_CNT reaches the value 0x36.	R/W
0x08	TIM_CLR	Clear register. If the TIM_CLR[0] bit is '1', the counter register (TIM_CNT) value will be set to 0. Other bits are ineffective.	R/W
0x0C	TIM_ENA	Enable register. If the TIM_EN[0] bit is '1', the counter will continue its normal operation. If the TIM_EN[0] bit is '0', the counter will retain its last value. If TIM_EN is '0' and TIM_CLR is '1', the counter register will be reset. Other bits are ineffective.	R/W
0x10	TIM_MOD	Mode register. If the TIM_MOD[0] bit is '1' it will count up, if it is '0' it will count down.	R/W
0x14	TIM_CNT	Timer counter register. The register where the counter is kept.	Read Only
0x18	TIM_EVN	Event register. The TIM_CNT register will increment by 1 each time it reaches the TIM_ARE value.	Read Only
0x1C	TIM_EVC	Event clear register. If the TIM_EVC[0] bit is '1', the event register (TIM_EVN) value will be set to 0. Other bits are ineffective.	R/W

UART

Offset	Name	Description	R/W
0x00	UART_CPB	Clock-per-bit register. It tells you how many clock pulses will be sampled for each bit in the UART packet. It is used to set the baud rate. For example, if there is a system clock at 48 MHz frequency, 48e6/9600 = 5000 must be written to the UART_CPB register to communicate with a	R/W

		baud rate of 9600 bps. UART baud rate will be calculated as (system clock frequency)/UART_CPB. The UART peripheral to be designed should work this way.	
0x04	UART_STP	Stop-bit register. According to the UART_STP[1:0] value, the stop bit width is determined as follows: “00” → Stop-bit 1 “01” → Stop bit 1.5 “1X” → Stop-bit 2 Other bits are ineffective.	R/W
0x08	UART_RDR	Read data register. It records 1 byte of data into UART_RDR[7:0] bits by the UART receiver. Other bits are ineffective. When data reception is completed, it sets the UART_CFG[1] bit to '1'.	Read Only
0x0C	UART_TDR	Transmit data register. It sends the data in the UART_TDR[7:0] bits to the other party when the UART_CFG[0] bit is set to '1'. When data sending is completed, it sets the UART_CFG[2] bit to '1'.	R/W
0x10	UART_CFG	Configuration register. Its features are as follows: - UART_CFG[0]: Transmit enable bit. As long as it is '1', it sends the data in the UART_TDR register. - UART_CFG[1]: Data received bit. When data arrives, it is set to '1' by HW. It must be set to '0' by SW. - UART_CFG[2]: Transmit completed bit. When the data in UART_TDR is sent, it is set to '1' by HW. It must be set to '0' by SW.	R/W

I2C Master

The SCL clock frequency of the I2C Master peripheral to communicate with the slave device will be at a fixed speed of 400 kHz. You can benefit from the documentation in the following github repo about the I2C Master working principle:

https://github.com/mbaykenar/apis_anatolia/tree/main/VHDL_ile_FPGA_PROGRAMLAMA/ders27

Offset	Name	Description	R/W
0x00	I2C_NBY	Number of bytes register. It refers to the number of bytes to be written to or read from the slave device by the I2C master peripheral. It can take a value between 1-4. If a value other than 1,2,3,4 is entered, it will be rounded to the nearest number. For example, if you write 0, it gets the value 1, if you write 25, it gets the value 4.	R/W
0x04	I2C_ADR	Slave address register. 7-bit mode is supported.	R/W

		I2C_ADR[6:0] bits define the slave address. Other bits are ineffective.	
0x08	I2C_RDR	Read data register. It contains the data to be read from the slave device by the I2C master. It can save 1 to 4 bytes according to the I2C_NBY register value. When the I2C_CFG[2] bit is '1', it starts the receive process. When the receive operation is completed, it sets the I2C_CFG[3] bit to '1'. The first arriving data is written into low significant bytes. For example, if I2C_NBY is 2, the first read byte is written to I2C_RDR[7:0] bits, and the second read byte is written to I2C_RDR[15:8] bits.	Read Only
0x0C	I2C_TDR	Transmit data register. It contains the data to be written to the slave device by the I2C master. It can transmit between 1 and 4 bytes according to the I2C_NBY register value. When the I2C_CFG[0] bit is '1', it performs the transfer. At the end of the transfer, it sets the I2C_CFG[1] bit to '1'. Transmission occurs starting from the low significant byte. For example, if I2C_NBY is 2, the data in I2C_TDR[7:0], then I2C_TDR[15:8] bits are transmitted.	R/W
0x10	I2C_CFG	Configuration register. Since transmit and receive transfer cannot be done at the same time in the I2C master, hardware precautions must be taken if receive and transmit are enabled at the same time while designing the peripheral. For example, if I2C_CFG[0] and I2C_CFG[2] are '1' at the same time, transmit is made or other measures can be taken. Its features are as follows: - I2C_CFG[0]: Transmit enable bit. As long as it is '1', it sends the data in the I2C_TDR register in the amount of I2C_NBY bytes. - I2C_CFG[1]: Transmit completed bit. When data transmission is completed, it is set to '1' by HW. It must be set to '0' by SW. - I2C_CFG[2]: Receive enable bit. As long as it is '1', it starts the transfer to read the number of bytes in the I2C_NBY register from the slave device. - I2C_CFG[3]: Receive completed bit. When reading data from the slave device is completed, it is set to '1' by HW. It must be set to '0' by SW.	R/W

QSPI Master

The QSPI Master peripheral will communicate with the non-volatile memory that will be used for the microcontroller program memory. An example NOR Flash (S25FL128S) datasheet with which the QSPI Master peripheral will communicate can be accessed from the link below:

https://www.infineon.com/dgdl/Infineon-S25FL128SS25FL256S_128_Mb_(16_MB)256_Mb_(32_MB)_3.0V_SPI_Flash_Memory-DataSheet-v20_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ecfb6a64a17

The QSPI Master to be designed will support x1, x2 and x4 data width. Data transactions will support 256 bytes page write and page read. There will be 3 byte addressing mode support, there will be no 4 byte addressing mode support, that is, a maximum of 128 Mbit flash will be supported. SPI mode 0 will be supported (SCLK pin will stop at '0' in IDLE state, CS pin will start sampling with the rising edge after falling to '0'). Dual-flash mode will not be supported. It will work as SDR (single data rate), DDR (double data rate) will not be supported. The following commands will be supported in QSPI Flash memory:

- READ
- DOR (Read Dual Out)
- QOR (Read Quad Out)
- PP (Page Program)
- QPP (Quad Page Program)
- SE (Sector Erase)
- READ_ID (Read Electronic Manufacturer Signature)
- RDID (Read ID – JEDEC Manufacturer ID and JEDEC CFI)
- RES (Read Electronic Signature)
- RDSR1 (Read Status Register-1)
- RDSR2 (Read Status Register-2)
- RDCR (Read Configuration Register)
- WRR (Write Register – Status-1, Configuration-1)
- WRDI (Write Disable)
- WREN (Write Enable)
- CLSR (Clear Status Register-1)
- RESET (Software Reset)

To send these instructions to flash memory, the appropriate instruction value will be written to the "Instruction" bits in the QSPI_CCR register. Instruction values should be adjustable software. For sample instruction values, the S25FL128S 128Mbit QSPI flash memory whose datasheet link is given above can be examined.

Offset	Name	Description	R/W
0x00	QSPI_CC	Communication configuration register. When a	R/W

	R	transfer occurs for writing or reading to flash memory, a value is written to this register. The QSPI peripheral controller initiates the transaction for reading/writing with the flash memory when a write is made to this register. - QSPI_CCR[7:0]: Instruction value. It contains the instruction value to be sent to flash memory. - QSPI_CCR[9:8]: Data mode: “00” Sending data “01”x1 “10”x2 “11”x4 - QSPI_CCR[10]: '0' Reading from flash memory '1' Writing to flash memory - QSPI_CCR[15:11]: Number of dummy cycles. The flash memory may request a dummy cycle with READ, DOR, QOR or other commands (the datasheet should be examined and a decision is made). In this case, a dummy cycle is applied as much as the value in this field. - QSPI_CCR[24:16]: Data size to be written or read to flash memory. One more byte of data is written or read than the value written into this field. For example, if the WRR command will only be written to Status Register-1 in the Flash memory, 0 is entered in this field, and if it is also written to the Configuration Register, 1 is entered. 255 is entered for the PP command. Reading is similar. - QSPI_CCR[30:25]: Prescaler value. The value of the counter register will change by dividing the clock frequency by one more than the value entered in this field. For example, if '0' is entered in this field, SCLK will be at the system clock speed. When this field is '1', SCLK will be at half the system clock speed. - QSPI_CCR[31]: Clear status register '0' No effect ‘1’ Clear status register QSPI_STA.	
0x04	QSPI_AD	Address register. It contains the address	R/W

	R	information to be read or written in flash memory. It is used as 3-byte in READ, DOR, QOR and PP command. That is, QSPI_ADR[23:0] bits are sent as the address.	
0x08	QSPI_DR 0	Data register. A page read or written from flash memory contains the least significant (LSB) 4 bytes of data, Page[31:0]. If a register is to be written to the flash memory with the WRR command, the value written to this register is written. The values to be written to the register are written using the least meaningful bytes in this register.	R/W
0x0C	QSPI_DR 1	Data register. A page to be read or written from flash memory contains 2.4 bytes of data, Page[63:32].	R/W
0x10	QSPI_DR 2	Data register. A page to be read or written from flash memory contains 3.4 bytes of data, Page[95:64].	R/W
0x14	QSPI_DR 3	Data register. A page read or written from flash memory contains 4.4 bytes of data, Page[127:96].	R/W
0x18	QSPI_DR 4	Data register. A page read or written from flash memory contains 5.4 bytes of data, Page[159:128].	R/W
0x1C	QSPI_DR 5	Data register. A page read or written from flash memory contains 6.4 bytes of data, Page[191:160].	R/W
0x20	QSPI_DR 6	Data register. A page read or written from flash memory contains 7.4 bytes of data, Page[223:192].	R/W
0x24	QSPI_DR 7	Data register. A page read or written from flash memory contains 8.4 bytes of data, Page[255:224].	R/W
0x28	QSPI_ST A	Status register. QSPI_STA[0]: When it is '1', the read or write transaction is completed. It is reset when '1' is written to the QSPI_CCR[8] bit. QSPI_STA[1]: '0' Not busy '1' Busy. In other words, it will be '1' as long as a transaction continues with the flash memory.	Read Only

USB Full-Speed Device (12 Mbps)

Registry definitions will not be made specifically for the USB Full-Speed Device peripheral. The peripheral to be designed must support USB FS 12 Mbps speed. It does not have to work as “OTG”, it only needs to perform the “Device” function, that is, there is no need to support “Host” features. The part that the contestants are responsible for designing is only the "digital design" part. Analog or physical requirements may be ignored.

For detailed information about the USB peripheral, you can examine the USB FS Device titles of the microcontrollers given below:

STM32L0 series microcontrollers “USB FS Device Interface”:

https://www.st.com/resource/en/reference_manual/rm0376-ultralowpower-stm32l0x2-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

MICROCHIP SAM4S series microcontrollers “40. “USB Device Port (UDP)”:

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11100-32-bit%20Cortex-M4-Microcontroller-SAM4S_Datasheet.pdf

NXP A14x/15x series microcontrollers “Chapter 35 USB Full Speed (FS) Device Controller”

https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/mcx-arm-cortex-m/mcx-a-series-microcontrollers/mcx-a14x-15x-mcus-with-arm-cortex-m33-scalable-device-options-low-power-and-intelligent-peripherals:MCX-A14X-A15X

(Access to the document requires NXP login)

An open-source repo written in Verilog and with the USB-CDC functionality required in the specification:

https://github.com/WangXuan95/FPGA-USB-Device

JTAG (Optional)

The JTAG port has been added to the specification and is optional in order to debug the CV32E40P processor core, which is the RISC-V based processor core of the microcontroller. The CV32E40P processor has a “debug” port and the processor can be debugged through this port. The processor's debug port can be accessed via a JTAG TAP controller within the microcontroller. Details are available in the links below.

CV32E40P processor core Debug & Trigger documentation page:

https://docs.openhwgroup.org/projects/cv32e40p-user-manual/en/latest/debug.html#debug-support

Open-source github repo for processor debug access with JTAG TAP implementation:

https://github.com/pulp-platform/riscv-dbg/tree/master

7TH YEAR

MİLLİ TEKNOLOJİ HAMLESİ

TEKNOFEST

AEROSPACE AND TECHNOLOGY FESTIVAL

32