PAPER

Development of SRAM-APB protocol interface and verification

To cite this article: V Karthikeyan *et al* 2023 *Eng. Res. Express* **5** 045017

Engineering Research Express

PAPER

Development of SRAM-APB protocol interface and verification

V Karthikeyan[1], K Balamurugan[2], Lakshmana Rao Namamula[3,*] and F Jeya Brindha[4]

[1] Department of ECE, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India
[2] Department of ECE, Swarnandhra College of Engineering and Technology, Narsapur, Andhra Pradesh, India
[3] Department of Information Technology, University of Sierra Leone, Sierra Leone
[4] Department of ECE, Kalasalingam Institute of Technology, Krishnankoil, Tamilnadu, India
[*] Author to whom any correspondence should be addressed.

E-mail: lakshmanarao09@gmail.com **and** lakshmanarao@ipam.edu.sl

Keywords: On-chip communication, APB protocol, Modules, SRAM, System verilog

RECEIVED
3 April 2023

REVISED
24 September 2023

ACCEPTED FOR PUBLICATION
3 October 2023

PUBLISHED
13 October 2023

Abstract

The purpose of this mechanism is to enhance the chip's internal connections and read/write memory capabilities. The Advanced Microcontroller Bus Architecture (AMBA) is one such shared bus that uses static random-access memory to achieve this goal. As a result, it's important to weigh a variety of design options before diving into the Verilog description. It's also important to remember that the system must be designed to accommodate a large number of interoperable modules and memories. The design, on the other hand, starts with fewer modules and a less complicated description and realisation that relies on memory access. ModelSim is used to simulate after the delay has been modelled in Verilog. Since the interface's static random-access memory uses an APB protocol, its performance may be tested at this stage. In addition, Questasim employs verification modules and System Verilog technologies to guarantee the system's operation. From the obtained results, the Direct Memory Access (DMA) with SRAM-APB outperforms the alternatives, particularly in frame transmission schemes, with a wire efficiency that is 1.4 times higher and a dynamic energy efficiency that is nearly twice as high as those of conventional configurations.

1. Introduction

Static random-access memory (SRAM) is a category of semiconductor storage that stores each bit using bistable triggering technology. The word 'static' distinguishes it from dynamic RAM, which should be updated regularly [1, 2]. SRAM memory has data persistence, however, it is unstable in the sense that information is absent if the power is off. Synchronous SRAM is quicker than asynchronous SRAM and requires a clock signal to authenticate its control commands, allowing the cache to function in coordination with the central processing unit (CPU) [3, 4].

SRAM differs from dynamic random-access memory (DRAM), which must be updated regularly [5, 6]. SRAM is both quicker and costlier than DRAM [7, 8]; therefore it is often utilized in CPU cache, whereas DRAM is utilized in the main memory of a computer. Cypress is the world leader in Synchronous (Sync) SRAM, with lead times of six weeks or less, 99 percent or greater on-time shipment, and legacy product support lasting up to 20 years, having supplied more than 2.7 billion aggregate units. Customers in numerous market categories, including networking, automobiles, consumer electronics, commercial, healthcare, aviation, and defence, choose Cypress as their SRAM supplier [9].

The Advanced High-Performance Bus (AHB), Advanced System Bus (ASB), and Advanced Peripheral Bus (APB) are the three aspects of AMBA bus design [10]. The AMBA, AHB, or ASB buses are high-accomplished, high-bandwidth buses. Consequently, the high-bandwidth SoC (System on Chip) RAM and memory interfacing, high-performance advanced RISC processor,as well asdirect memory access masterbus are all coupled to the AHB/ASB [11]. The AMBA-APB bus has limited bandwidth and efficiency. So peripheral gadgets such as UART, keyboard, clock, and Peripheral Input Output (PIO) modules, which require less bandwidth, are interconnected to the APB. Figure 1 depicts the fundamental AMBA bus architecture.

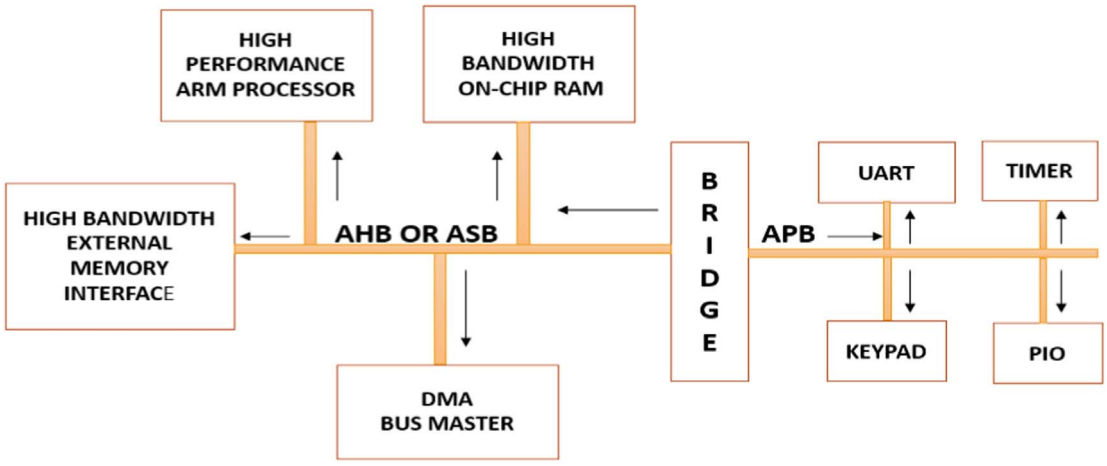

Figure 1. AMBA Bus Architecture.

The AHB or ASB connection links to the APB through this bridge. As a result, in APB, the bridge serves as the master, while all equipment linked to the APB bus serves as slaves. Interactions are initiated and transferred to the devices attached to the APB by a device on the high-performance links. As a result, the bridge is utilized to communicate between the high-performance bus and peripheral devices at any given time [12]. AMBA is utilised as an on-chip connector in such a design. It was once only used in microcontrollers, and it is already found in various ASIC and SoC elements, particularly executioncontrollers found in modern smartphones and tablets. AMBA is afreely available on-chip connectivity design for interconnecting and coordinating functional modules in a SoC. It aids in the construction of multi-processor architectures with a great numeral of controllers as well as peripherals, which means the first time it works correctly.

As an auxiliary bus to the main processor bus, the AMBA-APB offers a straightforward peripheral macro-cell interfacestructure. It's made up of memory-mapped registers that act as bridges [13]. The external storage contact is implementation-dependent and could only have a fine information path, but it can also facilitate a trial admittance approach that permits self-governing assessment sets to be employed to examine the core bus schemes independently [14]. When it comes to assisting peripheral functionalities, the APB is sometimes optimised to decrease communication difficulty as well as energy consumption. This bus can be implemented in combination with the processor bus edition [15].

APB has an unpipelined protocol. As a result, it interacts with minimum-bandwidth devices, which do not require the maximum response of the pipelined bus communication. Every data transition is related to the (+) edge of the clock, making it easy to incorporate APB peripherals in all design flows. APB may communicate with the AMBA AHB-Lite as well as the AMBA Advanced Extensible Interface (AXI). APB can be utilised for accessing the programmable control registers of peripheral gadgets [16].

All of the aforementioned buses linearly transmit information; however, data handling in some domains, including pattern recognition, imaging, and wireless networking, is typically relatedto the association of data neighbours, interaction, zones, borders, and prevents data load and store. Matrix or block transmissions of data are preferred to serial burst transmissions in this scenario. When most interactions stay within the identical bus level, the bandwidth is maximised, this is the case for some complex bus structures like multibus and layered designs. However, the additional expense in space and power is justified by the need for an extensive amount of lines and inside logic, like multiplexers for data transfer between layers and storage enabling data transfer management [17].

The paper aims to use System Verilog Hardware Description Language to integrate SRAM with APB protocol on-chip communication. The present integrated circuit technology design situation is distinguished by maximum throughput, sophisticated operability, and rapid time to market (TTM). To solve the above-mentioned issues, a reuse-incorporatedmethod for the SoCscheme has turned critical. The job entails the assessment of the APB Protocol and its slave SRAM. The aim is to place DUT in the assessment. The suggested template is utilised to link the master and slave (SRAM). The overall strategy was written in System Verilog and tested with Questasim.

The following are the work's highlights:

- Validation of the address and control is performed for every aspect of the un-pipelined access.
- Suggested strobe (un-clocked interface timing) interpretation saves the overall access time.

- The static peripheral bus interface consumes no power when the bus is inactive.
- Effectively write data implemented for comprehensive access using glitch-free.

2. Related works

The AMBA is utilised as aSoCbus in integrated circuit designs. It was previously utilised in microprocessor elements, but it is currently commonly utilised in a diverse variety of ASIC as well as SoC devices, particularly utilization mainframes employed in compact handheld wireless devices such as laptops and tablets. It simplifies the initially attempted construction of multi-processor platforms featuring a large number of ports, including accessories.

The Intellectual Property (IP) fundamentalof the APB busis created based on the AMBA 4.0 bus, and it transforms AXI4.0-lite connections (AXI-Master) to APB-4.0 connections (APB-Master). It provides a Slave interface and then executes them on the APB bus. AXI4 transactions are appropriately reduced to APB transactions because the APB protocol is substantially easier than AXI4 [18, 19]. A burst mode was described in detail by Prathibha and AmbikaSekhar [11]. An address transition detects (ATD) signal-producing circuit is described, which produces either an asynchronous ATD signal (a-ATD) or a synchronous ATD signal (s-ATD) based on the logic level of a mode signal (ATM). The disadvantage of this system is that it necessitates external components and clocks.

SandeepRaval [5] suggested a scheme Verilog-integratedvalidation of write functionality in SDRAM employing a memory regulator. With the increasing difficulty of contemporary technological systems and embedded system designs, validation is becoming critical to meeting the demands of shorter TTM. The most crucial part of the ASIC system design is validation. The validation work is predicted to occupy 40 to 70% of the total design process [20, 21]. For networking as well as other high-performance purposes, the QDR SRAM architecture offers the random memory access functionality required. Memory is a critical impediment in many applications for achieving improved system efficiency [9]. Kumar *et al* [15, 22] use Verilog Hardware Description Language (HDL) to develop an APB Bridge incorporating a clock skew minimising approach. The simulation software Vivado Design Suite ISim was utilised. The disadvantage is that clock-switching actions might result in significant power dissipation ranging from 15% to 50%.

Sanwatsarkar and Jagdish [23] discussed the construction of a DDR SDRAM processor utilising Verilog HDL, which provides an overview of the DDR simply by applying advanced circuit strategies to achieve higher throughput. DDR SDRAM (often referred to as DDR) sends information on both the positive and negative edges of the clock [24]. The DDR controller is built to provide suitable commands for SDRAM start-up, read/write access, periodic refreshing function, appropriate activation and precharge operations, and so on [25]. By sending data two times per cycle, DDR SDRAM increases the memory's bandwidth. The proposed DDR Controller has a 64-bit data range, a 4-bit burst length, and a 2 s Column Address Strobe (CAS)delay. The DDR assigns a synchronous knowledgeinterconnection as well as many control signals to the DDR SDRAM Memory [26]. This DDR SDRAM practical approach incorporates established deferrals and stage linkages throughout data sources and yields and ensures that tickers and delayed signals are free of bugs [9, 27].

The frequency of completely random read or write operations that memory may accomplish per second is known as the Random Transaction Rate (RTR). Mega Transactions per Second (MT/s) is the unit of measurement for RTR. In high-performance applications like networking, where memory access is inconsistent, RTR is an important statistic [28]. Reddy *et al* [29] provide an outline of the AMBA bus construction and a detailed explanation of the APB bridge. The APB bridge is intendedby means of the Verilog HDL protocol and validated by Xilinx. The suggested framework is utilized to interconnect the master and the slave for reliable information transmission. It is unable to interface with any high-bandwidth devices.

Kumar and Sinha [22] discussed the APB-3 protocol design and validation of the slave APB-3 practice. Attention examination is a critical component of the validation progression since it indicates how thoroughly the DUT's source code has been checked. The efficient attention examination improves validation efficacy and helps the validation expert identify untested functions.Static RAM that is used at a lesser speed, like in systems with modestly clocked microprocessors, consumes relatively minimal resources and can be as low as a few microwatts while idle. Several approaches to managing the energy consumption of SRAM-based memory architectures have been presented [30].

Even though these memories function well as primary memory for computing workloads, they may not satisfy the demands of high-performance applications which require a high RTR.To emphasise the significance of the proposed study, some previous SRAM works are provided below in table 1.

For master-slave behaviours, a shared bus such as AMBA (Advanced Microcontroller Bus Architecture) is proposed in this work. As a result, before beginning any Verilog specification, numerous design options are laid out and assessed. Additionally, this framework is built to support a large number of modules that can

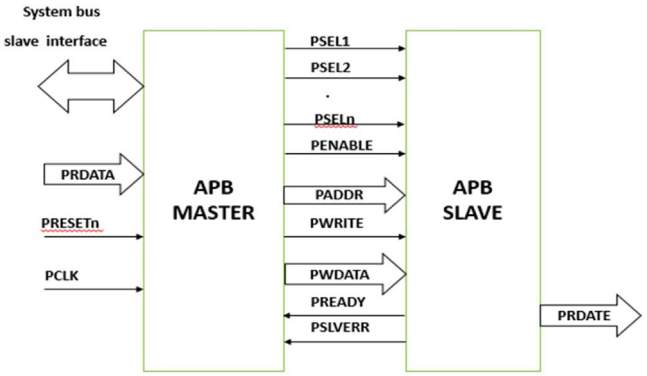

Figure 2. Block diagram of APB Protocol.

Table 1. Summary of literature works in SRAM.

Reference	Year	Inference
Calhoun and Chandrakasan [31]	2007	This research investigates the lower voltage operating constraints of conventional six-transistor (6T) SRAM and suggests a new type of bit-cell which can operate under substantially reduced voltages.
Ebrahimi *et al* [32]	2012	An alternative 8-T subthreshold SRAM unit with enhanced '0' write capability was suggested. Also proposed is an innovative kind of subthreshold SRAM unit with a 10-T subthreshold that uses less energy when idle and depends on FinFET architecture.
Pal and Islam [33]	2015	In this investigation, the authors provide an enduring asymmetrical SRAM unit and discuss the influence of varying the process settings on the recommended cell's different layout characteristics.
Gupta *et al* [34]	2017	The seven-transistor organism, which features noise-margin-free read functioning, has been presented as a solution to the read–write collision of the six-transistor unit.
Patel *et al* [35]	2018	Researchers introduced an SRAM unit that employs identical bit lines with half-select-free approaches to reliably produce high yields from nine-carbon nanotube transistors with field effects.
Pal *et al* [17]	2020	In this investigation, the author presented the construction of a radiation-hardened SRAM unit with the strongest QC of any cell under consideration, making it extremely resistant to soft errors.
Sachdeva and Tomar [36]	2021	A unique 9-T SRAM cell has been presented as a remedy to the read/write inconsistency of SRAM units, exhibiting less energy usage and modest delay. The new circuit design cuts down on power consumption by 74%–81% by minimising leakage.

communicate with one another. On the other hand, the approach begins with a smaller number of modules (eight) and a simpler description and execution. Following that, and after the eight-module system has been modelled, the model is expanded to N modules, where N can be any number. Finally, the simulation is performed via ModelSim once the delay has been modelled in Verilog. The system's behaviour is examined at this level. Furthermore, the system's behaviour is validated by utilising Questasim's System Verilog technology validation modules.

3. Materials and methods

The Advanced Peripheral Bus (APB) was developed by using the design specifications. Figure 2 shows the AMBA APB's basic block diagram, which illustrates the fundamental interaction signals.

PRDATA, PRESETn, and PCLK are input control signals for the APB Master, and PSELn, PENABLE, PADDR, PWRITE, PWDATA, PREADY, and PSLVERR are output control signals. The APB slave accepts PCLK, PRESET, PSEL, PENABLE, and PWRITE as source control inputswith PADDR, and PWDATA as 32-bit bridge inputs as well as outputs 32-bit PRDATA. Caches as well as other applications involving burst transmissions, up to 144Mbit per chip, are typically employed.

An SRAM with p-bit address buses and q-bit data buses has a capacity of 2p words or 2pq bits. The general wordlength is 1 byte, which means that 8bits can be written or read to all of the SRAM cells 2p distinct words.

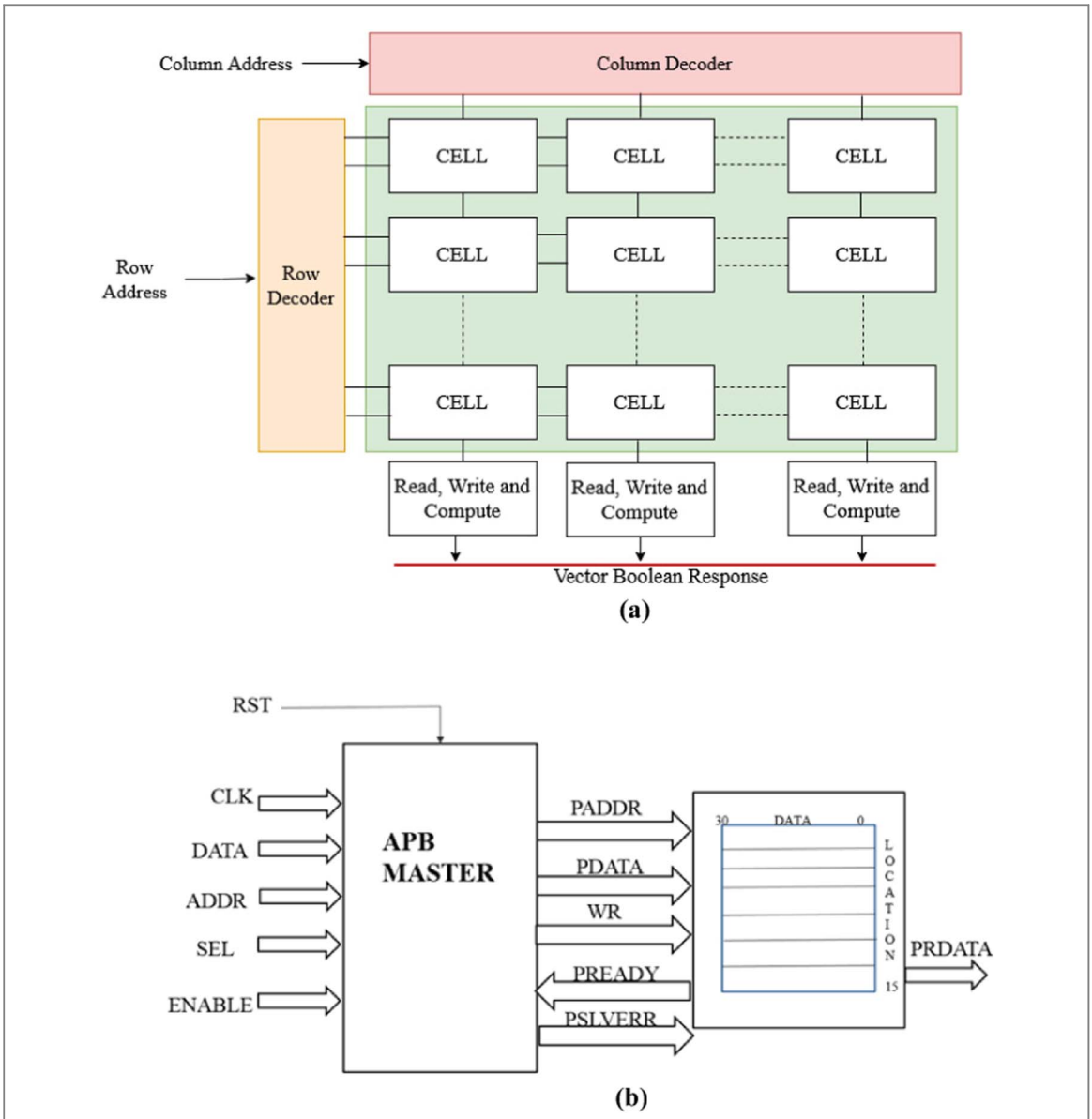

Figure 3. (a) Conventional SRAM array, (b). Interface SRAM with APB protocol.

Several typical SRAM ICs feature 11 address lines (therefore a size of 2p = 2,048 = 3d words) and one-byte words, referring to them as '2k 8 SRAM.'

3.1. Interface SRAM with APB Protocol

Figure 3(a) represents the conventional SRAM architecture [37] and 3(b) illustrates the APB Protocol's interaction with SRAM as well as its functions. As the slave component of the APB Protocol, the PADDR, PDATA, WR, and PREADY are the SRAM's inputs, while the PRDATA is the SRAM's output. The SRAM receives its information from the APB Master's outcome. The information will be stored in the address during the read operation. The information is transferred to PDATA. The data remains in the same position during the write process. PREADY is high when data becomes available for transport. The PREADY is low if it is not ready to transfer information.

The inverters in the static random access memory cell dynamicallyactivate the bit lines (BL) one and zero throughout read operations. In a DRAM, the BL is coupled to storage space units; therefore, charge sharing leads the BL to shift up or down. This enhances SRAM bandwidth (BW) compared to DRAMs. SRAMs' symmetric form also permits differential signalling, which makes minor voltage fluctuations easier to detect [38].

3.2. Operation States

An SRAM cell can be in one of three situations: standby (idle), read (data request), and write (update data). Readability and write stability should be present in SRAM in both read and write modes. The following is how the three states work:The access transistors cut off the cell from the BLs in standby mode but the word line (WL)

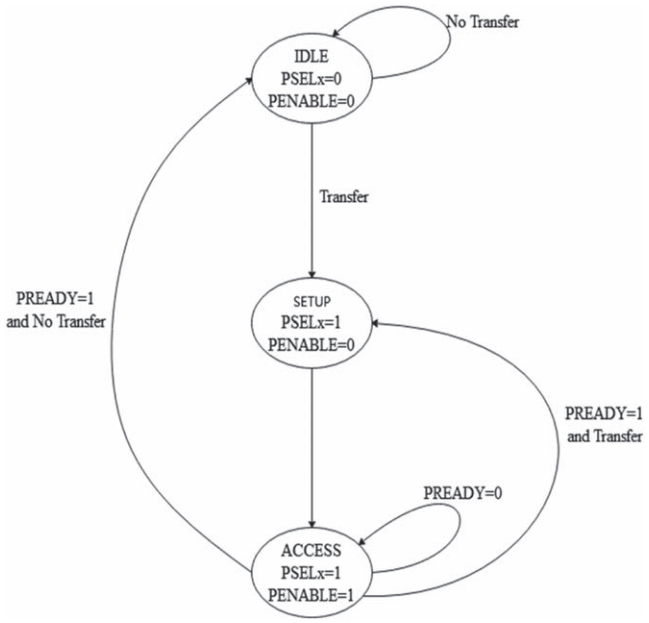

Figure 4. Operation states.

is not activated. Since they are attached to the power source, 2 cross-coupled inverters (CCI) will keep reinforcing one another.

Reading requires merely declaring the WL as well asreading the SRAM unitcondition using a distinct contact transistor and the BL in the reading operation. BLs are quite lengthy and have a significant scrounging capacitance. Both BLs are precharged to the maximum (L-1) level before the read cycle begins. The WL is then emphasized, enabling mutual accesses and causing the voltage on one-bit line BL to drop significantly. A logic amplifier detects which line has the stronger voltage and consequently determines whether a 1 or a 0 was accumulated. The read operation is faster with higher responsiveness of the logic amplifier. Step-down is simple with the nMOS since it is stronger. As a consequence, BL is usually precharged with a maximum voltage. Various scientists are also attempting to minimizepower consumption by pre-charging at a low voltage.

The write cycle starts with the data to be written being applied to the BLs. If we intend to write a zero on the BLs, we'll deposit one BL to one and another BL to zero. The BL's contents are upturned to create a '1'. Then, WL is emphasized, and the data to be accumulated is latched as input. This works since the BL input drivers are configured to be much stronger than the cell's relatively fragile transistors, permitting them to simply conquer the CCI prior state. Entrance nMOS transistors should be more powerful than lower nMOS or uppermost transistors to be effective. As a result, when the write process slightly overrides one transistor pair, the gate voltage of the opposing transistor pair changes as well.

RAM having an approach duration of 60 ns will respond with valid data within 60 ns of the address lines becoming genuine in bus behaviour. However, the data will be retained for a delay period of 5–8 ns. Fall and rise times have a 5 ns effect on valid time slots. Reading the lower part of an address range, bits in sequence allow for much faster access time (25 ns).

Figure 4 illustrates the operation states of the SRAM. The default phase is IDLE, which means no function is being executed. The PSEL signal is asserted to initiatethe setup phase. When data exchange is necessary, the bus initiates the setup state. Throughout this process, the PWRITE, PADDR, and PWDATA commands are available. The bridge will reside in the setup phase for a single clock cycle formerlytransferring into the access phase on the subsequentpositive edge of the clock. The start of the ACCESS stage is signalled by the affirmation of the PENABLE input.

Through the transition from the setup stage to the access stage, all control, as well as data signals remain steady. During this step of the read process, PRDATA is available on the bus. For one clock cycle, the PENABLE signal is likewise high. The bus will shift to the IDLE phase if no more information is needed. However, if more information exchange is needed, the bus may enter the SETUP stage.

3.3. Verification

As shown in figure 5, one of the most critical phases of the VLSI design process is verification. Its goal is to detect faults in the RTL (Register Transfer Level) design trial such that they don't turn out to be damaging later on in the

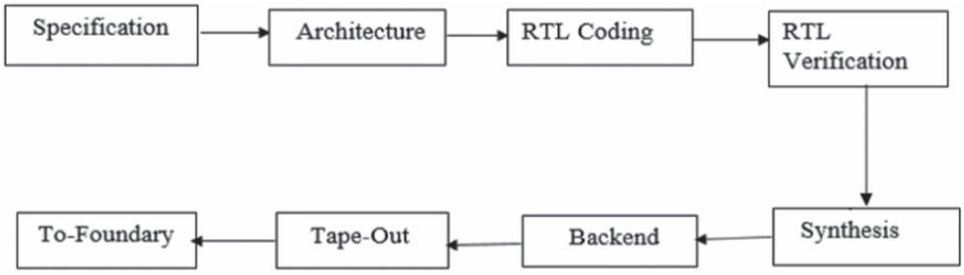

Figure 5. Verification flow diagram.

Table 2. Verification class description.

Class	Explanation
Transaction	Defines whether the pin level activity is generated by the agent or whether the activity must be noticed by the agent.
Generator	The stimulus (transaction) is generated by the generator and the packet-level data within the transaction is driven into the pin level.
Monitor	Observes pin-level activity on interface signals and translate it to packet-level data, which is then delivered to components such as the scoreboard.
Agent	The container class (agent) groups the classes that are particular to an interface or practice.
Score Board	Receives data from monitors and compares it to the values expected. Expected values might be created from a reference model or a golden reference value.
Environment	The agents and indicators are only two examples of more advanced elements that can be contained in the setting module.
Test	• Testing is responsible for configuring the testbench. • Begin the process of creating testbench components. • Initiate the stimulation drive.
Testbench top	This is the class that interfaces the DUT to the Testbench at the top. It comprises DUT, Test, and interface instances, with the interface serving as a link between the DUT and the Testbench.
Mailbox	Messages are exchanged between processes via a mailbox. A mailbox is a simple way to communicate between processes. With the put() and get() methods, we may send and receive messages.

system design. The complexity of integrated circuits (ICs) has increased as the number of transistors has grown, feature sizes have shrunk, and design tools have improved. This increases the likelihood of bugs in the design appearing. As a result, the requirement for IC verification arose [41, 42].

The Verification Environment, also known as the Test Bench, is used to verify the proper operation of the design under test (DUT) by producing and applying specified input data into the model, capturing the design output, and comparing it to the desired outcome [43]. The verification environment is created by combining the various components that conduct certain tasks or operations. Various classes will be built in the verification environment to execute certain tasks, such as creating stimuli, driving, tracking, and so on, and each class will be named after the task [44].

The following are the phases involved in building the verification environment: formulation of a verification plan, testbench structure, and test bench writing. The set of classes that the APB verification environment is shown in table 2.

4. Result and discussion

The system's behaviour is examined in this section. Furthermore, the system's behaviour is validated using verification components built with System Verilog technology in Questasim.

4.1. Write operation in SRAM Cell

For a '0' to be properly written to a bitcell, the entry nMOS must be large enough to defeat the pMOS within the cell in the ratioed logic. To complete a write, the bitcell has to turn one shot, which forces its inner potentials to the proper levels. The write is unsuccessful, and the SNM is affirmative on the butterfly chart if the unit maintains bistability. If the SNM is adverse, then the write was effective (the cell is monostable). Device size is effective at keeping the right balance of currents for writing the SRAM unit for above-operational nMOS transistors (owing to mobility) and comparatively low reliance on [17, 31].

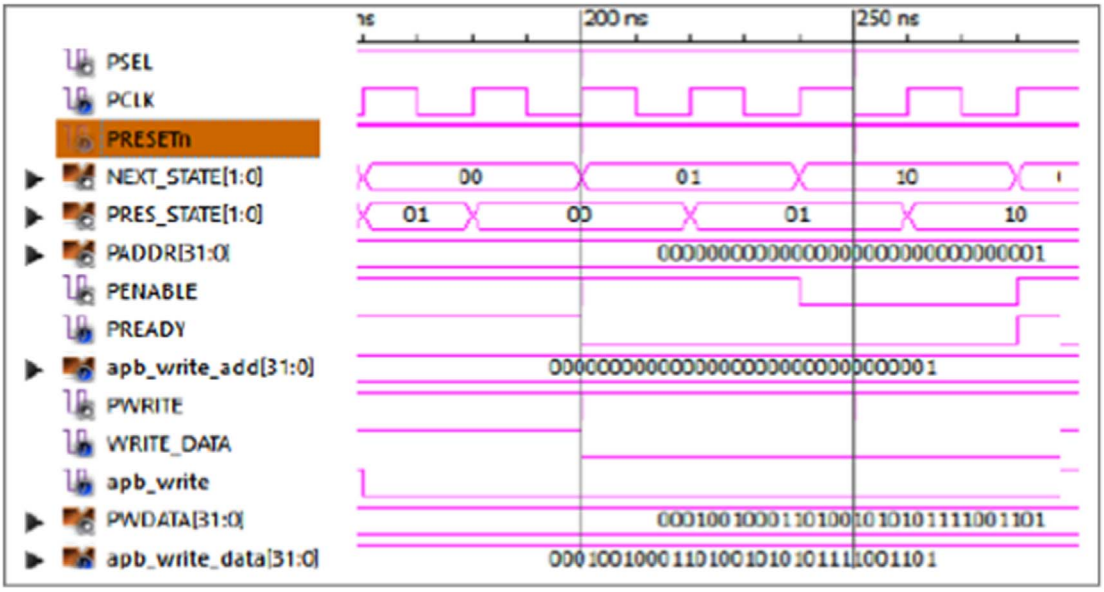

Figure 6. Simulation result for write cycle.

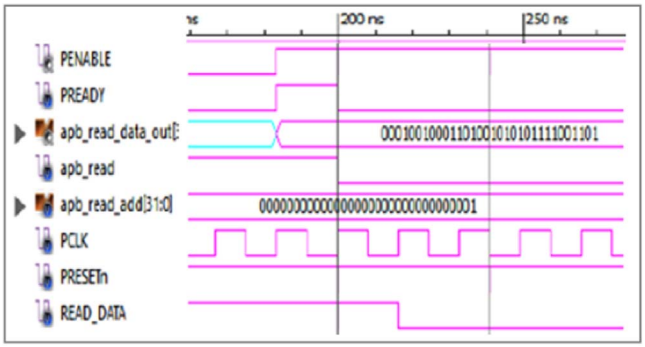

Figure 7. Simulation result for read cycle.

Figure 6 depicts the system's response to a write operation. When an operator's clock signal goes high, PENABLE and PREADY both go high, as does PADDR, which is 32 bits long. After enabling PADDR, data will be taken and written to PWDATA, as well as APB_write data, both of which are 32-bit in size.

4.2. Read operation in SRAM cell

Each node inside a bitcell is a potential introduction point for valuable sources of interference. When the value grows, the cell becomes less stable. When the cell's temperature goes above the SNM, it stops being bistable and loses its information. SRAM performance is more severely constrained by the consistency of its cells during active use compared to when it is on pause. The wordline as well as bitlines are initially loaded to a '1' state at the start of a reading session. As the signal voltage divides between the entry transistor and the pilot transistor, the inner node of the bitcell representing a zero is dragged upwards by the entry transistor, degrading the Read scenario in the SRAM unit [17, 32]. Figure 7 depicts the read operation performance of the suggested design. After transferring the data to the address and toggling once with PENABLE, APB_read goes high and reads the data from the APB_write_data and also stores it in the APB_read_data.

4.3. Simulation result of APB protocol

APB transfer error circumstances are identified by PSLVERR. Both read and write transactions can have error conditions. PSLVERR is considered legitimate once PSEL, PENABLE, as well as PREADY, are all HIGH through the final cycle of an APB transmission. As soon as PSLVERR is not being tested, it is directed, but not essential, that you move it low. That is, when any of the following values are low: PSEL, PENABLE, or PREADY.

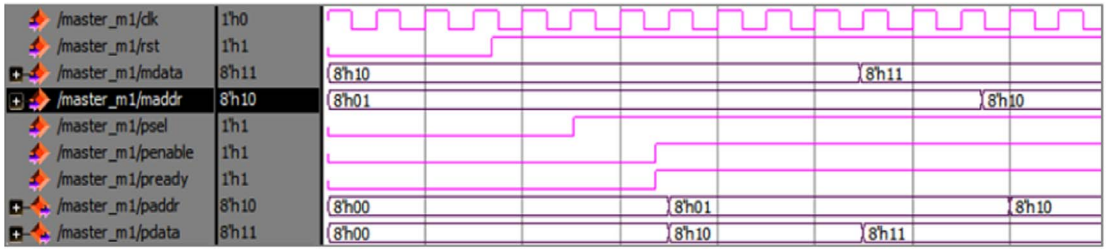

Figure 8. Master of APB.

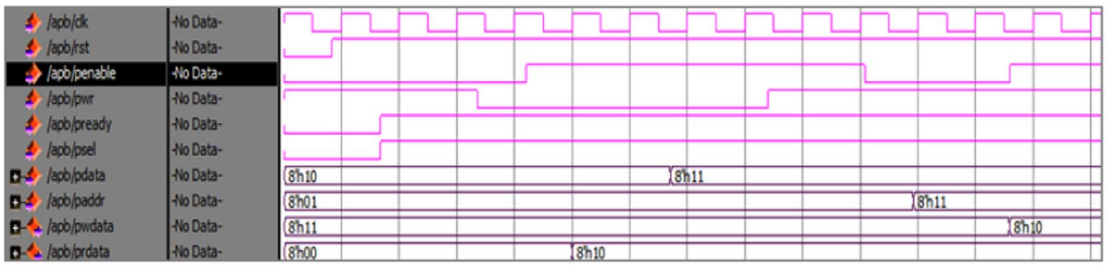

Figure 9. Slave of APB.

Transactions that fail may or may not have changed the peripheral's state. This is a peripheral issue, and any option is fine. When a write transaction fails, it doesn't indicate that the peripheral's register has not been updated. If a read transaction fails, the data returned may be incorrect. For a read mistake, the peripheral does not have to activate the DB to all 0s. The PSLVERR pin is not required by APB subunits. This is accurate for all the present and recently intended APB peripherals. Whenever a device fails to contain that pin, the APB bridge's appropriate input will be tied to the ground.

The PSEL, PWRITE, PADDR, and PWDATA inputs are declared at the T1 clockedge during the SETUP phase, which is shown in figure 8. The PENABLE and PREADY signals are asserted when the clock T2 rises to the next rising edge. The 'ACCESS cycle' is a term used to describe this process. If more data transfer is needed, the PENABLE signal is deactivated at the clock edge T3 and the PREADY signal transitions from high to low. In the master's reading functioning, the PSEL, PENABLE, PWRITE, as well as PADDR impulses are activated on the first phase clocking edge (SETUP cycle). PENABLE, PREADY, and PRDATA are asserted at the clock edge of T2 (ACCESS cycle), also PRDATA is read in this stage.In figure 9, the slaves APB's read and write operations are depicted.

When an operator's clock signal goes high, PENABLE and PREADY both go high, as does PADDR, which is 32 bits long. After enabling PADDR, data will be taken and written to PWDATA, as well as transmitted to APB_write_data, all of which are 32–bit in size, as shown in figure 9.

4.4. Simulation results of interface SRAM

SRAM is governed by bus commands, which are created by combining the EN, PSEL, and WE signals. When all three signals are high on a clock cycle, the accompanying command is a 'No Operation' (NOP). When the chip choice is not asserted, a NOP is also displayed. To maintain the integrity of the data held in SRAM devices, they must be refreshed regularly. The Auto Refresh command is issued by the SRAM Controller Core regularly. There is no need for the user to get involved.

4.4.1. Interface SRAM

On the APB, there is only one bus master, so an arbiter isn't required, as illustrated in figure 10. The master controls the address and writes buses, as well as performing combinatorial address decodes to determine which PSELx signal to activate. It's also in charge of timing the transfer by driving the PENABLE signal. During read transfers, it transfers APB data to the system bus.

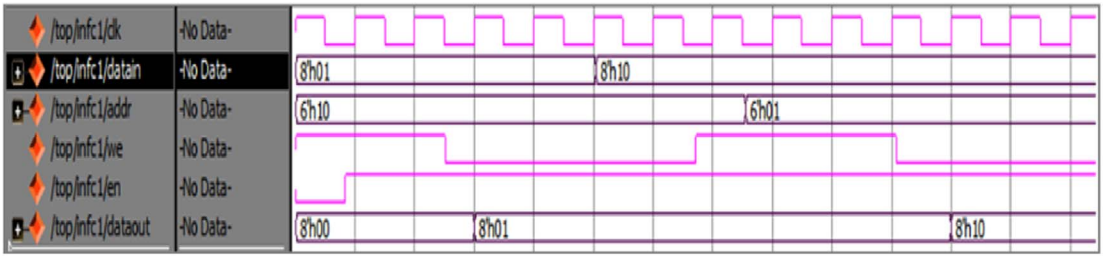

Figure 10. Interface SRAM.

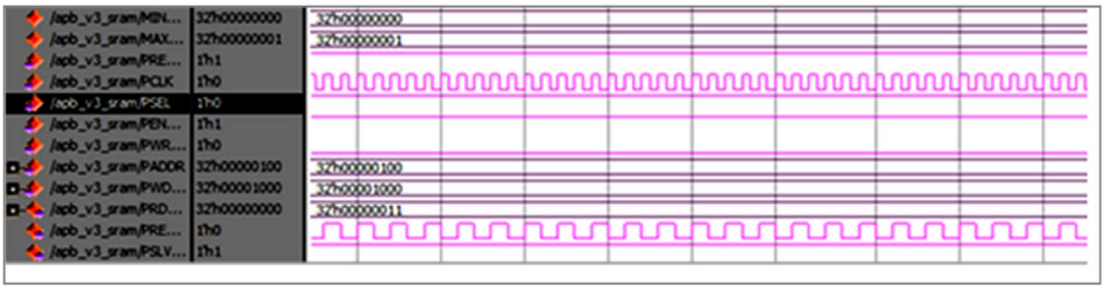

Figure 11. Simulation of 8-bit interface SRAM with APB Protocol.

4.4.2. Interface SRAM with APB protocol

Four banks are common in SRAM devices. Before any addresses may be written to or read from, these banks must be unlocked. With the active command, enrollment has been completed for both the opening row and bank. A bank may need to be closed and reopened to a subsequent row before it can be read from or written to. A precharge signal is used to close a bank account. Because enrolling and finishing banks consume memory BW, the SDRAM core was created to continuously control and accomplishes the position of all 4 memory banks. In this way, the control systems can strategically open and close banks. Figure 11 shows the simulation result of an 8-module SRAM interconnects using the APB protocol. Following the assessment of the system's performance for 8 modules, the technique is implemented for N modules, with N = 16 in this case, as shown in figure 12.

After transferring the data to the address and readying it with PENABLE, APB_ read goes high and receives the data from the APB_write_data and places it on the APB_read_data. The user experience of an APB slave is simple but can be customised. The design flow chosen will decide the precise implementation of the interface, and several options exist. These two signals, PSLVERR and PREADY, are the most important for preventing loss of data while data is being sent.

Figure 12 depicts the 16-bit data stored in the memory [size], as well as the 16-bit addressing. When the PREADY goes high, the PSLVERR rises high as well, indicating where the problem occurred in the programme. Figures 13 and 14 show the object as well as the transcript box parameters when interacting with SRAM.

4.5. Verification and performance evaluation

Effective simulation studies must be performed in the conceptual stage to promote development and forecast the dynamic system functionality of multi-function, multi-application SoC designs. The process for evaluating performance starts with algorithm analysis, and then System Verilog hardware verification language models several performance metrics [39]. During front-end simulation, two output files are produced: one is a VCD document containing input exchanging behaviours, and the second is a QOS document containing transferral delay and acceptable BW. It is possible to obtain a hardware expense file during the synthesis process that includes the maximum functioning BW, the quantity of IOs, as well as the number of slice registers and LUTs. Additionally, the comprehensive energy usage data can be created in a PWR file by feeding the correctly routed and directed NCD file, the PCF file regarding physical restrictions, and the VCD file for a particular simulation into the X-power analysis programme.The proposed bus performance assessment is generated automatically after the energy estimating framework, programmed in Perl and TCL scripts, computes the power and efficacy measures. The determined bus performance indicators, like time taken, wire efficacy, usable information BW, and energy efficiency (EE), are listed below.

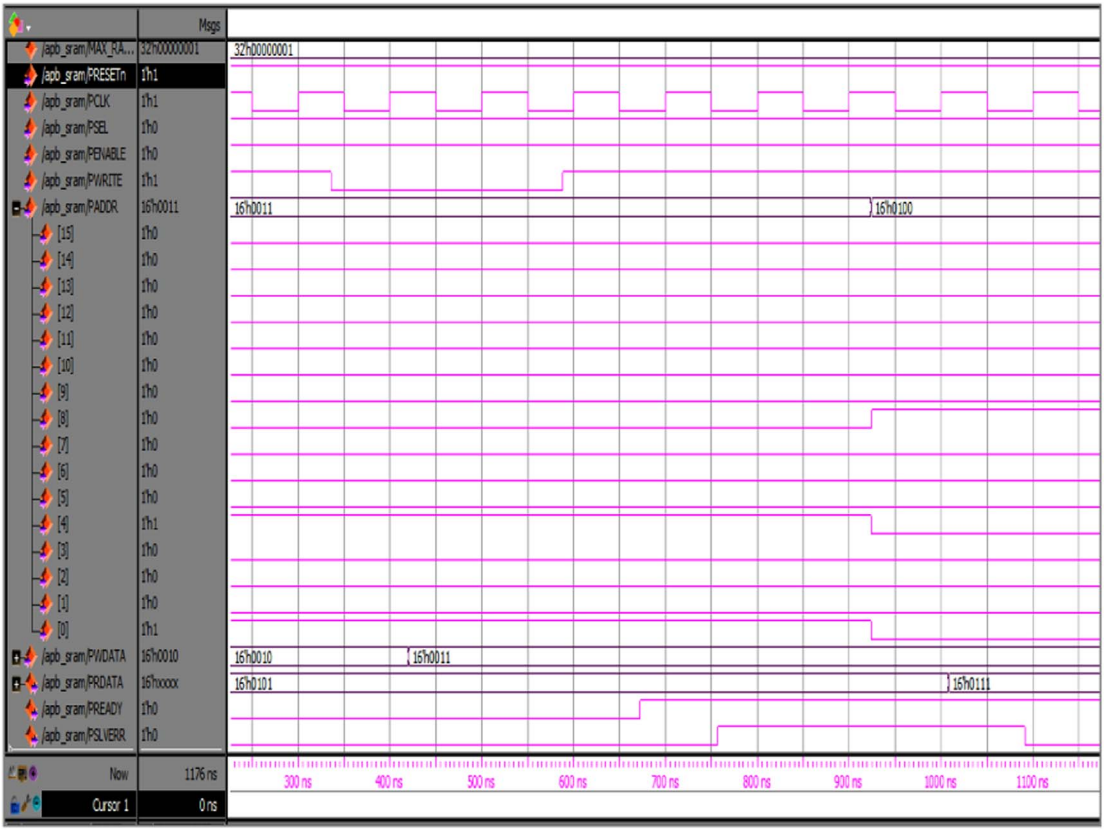

Figure 12. Simulation of 16-bit Interface SRAM with APB Protocol.

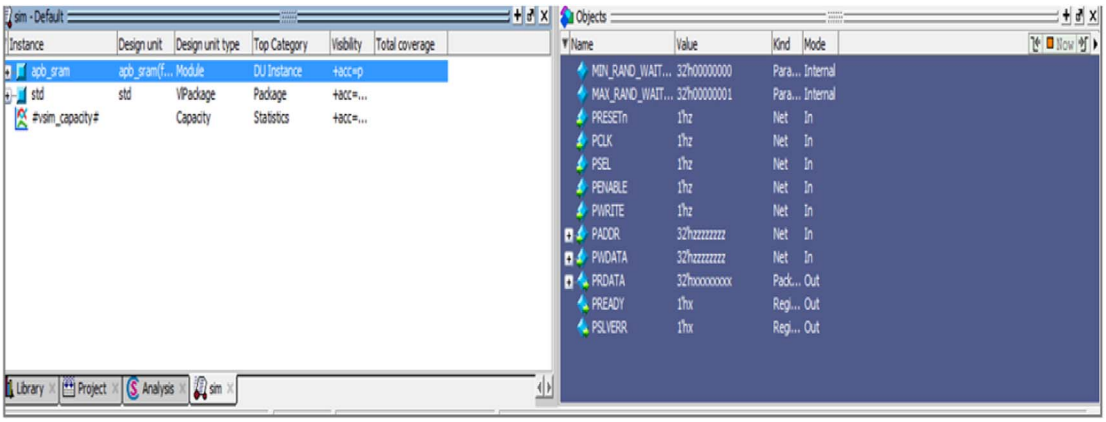

Figure 13. Object box of interface SRAM.

4.5.1. Transfer latency (TL)

The amount of transfer time consumed is calculated by multiplying the total amount of information transition period by the clock cycle, or the inverse of the clock frequency represented henceforth by f. Assuming that every bus transition always results in a quick response is necessary to concentrate on the bus layout. Along with this bus request, grant, and address transactions, two back-to-back transactions may also be combined [40]. Let Q stand for the likelihood of pipelined transfers, and let M and IB stand for the quantity of data as well as the amount of the information burst, etc. The TC_{HL}-indicated APB linear transfer delay is

$$TC_{HL} = \frac{\left\{\left[3 \times ceil\left(\frac{M}{IB}\right) + M\right] - 3Q \times ceil\left(\frac{M}{IB}\right)\right\}}{f} \quad (1)$$

Figure 14. Transcript Box for Interface SRAM.

Let M(dl) and Q(dl) be the data quantity and pipelined transferal likelihood of every DL, respectively, and HT signify the block height. As a result, the APB block transfer communication time denoted by TC_{HB} is

$$TC_{HB} = \frac{\left\langle \sum_{dl=1}^{HT} \left\{ \left[3 \times ceil\left(\frac{M(dl)}{IB}\right) + M(dl) \right] - 3Q(dl) \times ceil\left(\frac{M(dl)}{IB}\right) \right\} \right\rangle}{f} \tag{2}$$

As a result, the AXI linear TL indicated through $\boldsymbol{TC_{XL}}$ is

$$TC_{XL} = \frac{\left\langle \max(PO_{rd}, PO_{wr}) \times \left\{ \left[3 \times ceil\left(\frac{M}{YB}\right) + M \right] - 3Q \times ceil\left(\frac{M}{YB}\right) \right\} \right\rangle}{f} \tag{3}$$

Where PO_{rd} andPO_{wr}, indicates the probabilities of read as well as write. Similarly, the TL of APB bridge transmissions indicated by means of TC_{XB} is

$$TC_{XB} = \frac{\left\langle \max(PO_{rd}, PO_{wr}) \times \sum_{dl=1}^{YT} \left\{ \left[3 \times ceil\left(\frac{M(dl)}{YB}\right) + M(dl) \right] - 3Q(dl) \times ceil\left(\frac{M(dl)}{YB}\right) \right\} \right\rangle}{f} \tag{4}$$

Where YT represents the height of the APB block.

4.5.2. Wire efficiency (WE)

The mean ratio of data transfers per clock cycle to the total numeral of bus wires is referred to as wire efficacy as a bus-efficiency statistic [40]. X stands for the fundamental wire number. The calculation of WE is as follows:

$$WireEfficiency = \frac{\left\{ \frac{M}{(TC \times f)} \right\}}{X} \tag{5}$$

4.5.3. Valid data bandwidth (VDB)

Assuming that the bus widths (BW) and frequencies of the AHB, AXI, and APB buses areidentical, the typical bandwidth of the APB (BW_P) and AXI (BW_X) buses is twice that of the AHB bus (BW_H). All of these are written as follows:

$$BW_H = f \times BW \tag{6}$$

Table 3. Comparison of resources.

BUS type	Resource constraints			
	IOs	Slice registers	Slice LUTs	MOF (MHz)
AHB	381	11658	26214	368.5246
AXI	562	12894	17982	252.5785
Proposed	296	10745	16876	278.1050

Table 4. Comparison of power consumption of various bus configurations.

Test cases for the buses	Bus configuration	Power consumption		
		SP(mW)	DP(mW)	TP(mW)
HL	AHB	1238	564	1802
	AXI	1198	502	1700
	Proposed	1074	206	1280
XL	AHB	1194	668	1862
	AXI	1185	615	1800
	Proposed	1079	421	1500
IL	AHB	1202	465	1667
	AXI	1197	398	1595
	Proposed	1078	344	1422
HB	AHB	1230	356	1586
	AXI	1214	279	1493
	Proposed	1071	213	1284
XB	AHB	1286	558	1844
	AXI	1197	516	1713
	Proposed	1085	428	1513
IB	AHB	1312	502	1814
	AXI	1208	483	1691
	Proposed	1075	386	1461

$$BW_P = BW_X = f \times 2 \times BW \tag{7}$$

The definition of valid data bandwidth (VDB) is the amount of data that can be sent in a single cycle while still being valid.

$$Validdatabandwidth(VDB_H) = BW_H \times \frac{M}{TC_H \times f} \tag{8}$$

$$Validdatabandwidth(VDB_{PorX}) = BW_{PorX} \times \frac{M}{TC_{PorX} \times 2 \times f} \tag{9}$$

4.5.4. Dynamic energy (DE) and dynamic energy efficiency (DEE)

Power is essentially made up of energy. Dynamic energy consumption can be described as Let P_{dy} be the dynamic power with T signifying the entire transmission duration.

$$Dynamic\ \ Energy\,(DE) = \int_0^T P_{dy}\,dt \tag{10}$$

If $\boldsymbol{P_{dy}}$ is the uniform or mean power in this equation, then the DE utilization can be approximated as the combination of mean power and transmissionperiod. In addition to conventional power and energy assessment, we introduced a new metric, DEE, in our studies to analyse the association between effective data rate and energyutilization, which is expressed as:

$$Dynamic\ \ Energy\ \ Efficiency\,(DEE) = \frac{Valid\ \ Data\ \ Bandwidth}{P_{av}} \tag{11}$$

P_{av} denotes the mean dynamic power. DEE measures energy utilisation concerning effective information volume transmitted per sec per watt or effective informationvolume transferred per joule. The resource utilization of the conventional and proposed bus system comparison is represented in table 3.

The static power (SP) is essentially consistent for various test scenarios for a specific NCD file because it is primarily generated at the circuit level. So, the main goal of our study is to examine dynamic power (DP).

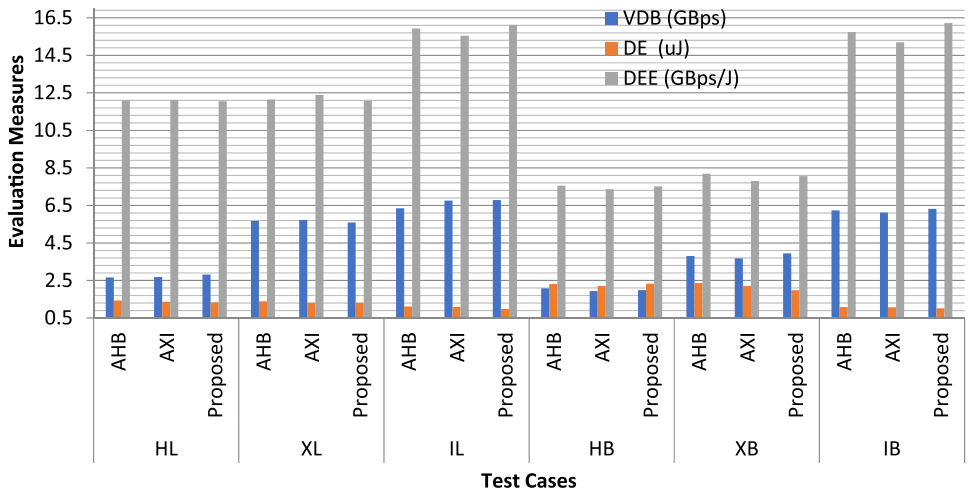

Figure 15. Performance of the proposed scheme against various test cases.

Table 5. Indices of Competence in Experiments.

Used case for testing the buses	Bus configuration	Experimental performance metrics				
		TC (ns)	WE	VDB (GBps)	DE (uJ)	DEE (GBps/J)
HL	AHB	6124	0.0052	2.66	1.43	12.10
	AXI	6116	0.0046	2.69	1.37	12.12
	Proposed	6042	0.0038	2.82	1.34	12.05
XL	AHB	2925	0.0054	5.67	1.39	12.14
	AXI	2854	0.0043	5.73	1.31	12.39
	Proposed	2948	0.0040	5.59	1.31	12.08
IL	AHB	2610	0.0087	6.35	1.12	15.93
	AXI	2688	0.0081	6.76	1.09	15.54
	Proposed	2597	0.0084	6.79	0.98	16.12
HB	AHB	9235	0.0045	2.08	2.32	7.55
	AXI	8956	0.0036	1.93	2.21	7.37
	Proposed	8984	0.0033	1.99	2.33	7.52
XB	AHB	4261	0.0041	3.81	2.37	8.19
	AXI	4274	0.0037	3.68	2.21	7.80
	Proposed	3862	0.0038	3.94	1.98	8.06
IB	AHB	2932	0.0079	6.23	1.08	15.74
	AXI	2759	0.0081	6.12	1.07	15.19
	Proposed	2606	0.0082	6.32	1.02	16.22

TC-Transfer Latency; VDB-Valid Data Bandwidth; WE-Wire Efficiency; DE: Dynamic Energy, DEE: Dynamic Energy Efficiency.

To satisfy the significantly increased switching actions of IOs, internal logic, and inputs, the block scenario uses more dynamic power than the directrelation for the equivalent NCD, as indicated in the 3rd column of table 4. APB-BUS uses less DP than AXI in each situation. The DP of APB-BUS is significantly higher than that of AHB due to its density and high throughput transfer properties, though. APB-BUS consistently uses less DP than normal buses. However, the proposed configuration's DEE is much higher than that of cutting-edge bus schemes because of its compactness and extraordinary data rate qualities. The Experimental Performance Metrics for various test cases of buses are shown in table 5.

This research proposes a high-performance bus with an easy-to-use interface, low power requirements, and fast data transfer rates. In addition, four empirical metrics (TC, WE, VDB, and DEE) are used to accurately and automatically evaluate its performance throughout the hardware implementation cycle. The results of a case study that compared the DMA (direct memory access) implementations of AHB, AXI, and APB-BUS show that the proposed method works better, especially for frame transmission schemes: the WE of APB-BUS is 1.2 times greater than AXI as well as 1.5 times greater than AHB, and the DEE is almost twice as high as that of AHB or AXI. The reduced power consumption and simplified circuitry are two benefits of APB-BUS's single-processor, multiple-client bus architecture. To ensure the accuracy of the performance models, we look into and evaluate

every metric of the three available bus protocols. Figure 15 displays all the information gained from the computational simulations.

Based on the findings, the suggested bus configuration may double the throughput of AHB and AXI while maintaining the same power and time requirements during block transmissions.

5. Conclusion

In this research work, the AMBA bus layout is taken as the base element, and the interface of APB to SRAM is investigated in depth. System Verilog verification is used to validate the APB after it has been constructed and interfaced with SRAM using Verilog HDL. Eight modules' results from simulations show that data transmission from a certain memory location may be traced back to the location where it was originally recorded. This results in an architecture that is both computationally efficient and more sound from a functional perspective. The effectiveness of the N module is then assessed, and obtained better result than the conventional bus configurations. Improved performance was attained with very low power consumption using the proposed approaches. The System Verilog Report Summary guarantees the design functionality as well. High TC, WE, VDB, DE, and DEE evaluation metrics show that SRAM-APB combinations outperform AHB and AXI buses by 8% and 5%, respectively. As the difficulty of designing electronic systems continues to rise, a technological representation of the same layout will inevitably be developed. Verilog is used throughout the development of the model for the APB-SRAM interface, and the model is validated on a System Verilog test bench.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Conflict of Interest declaration

The authors state that they have no financial ties to, or participation in, any organisation or entity that would benefit financially from the publication of this paper or its contents.

ORCID iDs

V Karthikeyan https://orcid.org/0000-0003-2974-6554
K Balamurugan https://orcid.org/0000-0001-5226-1665
Lakshmana Rao Namamula https://orcid.org/0000-0001-5617-3339

References

[1] Abbasian E and Sofimowloodi S 2023 Energy-efficient single-ended read/write 10t near-threshold sram *IEEE Trans. Circuits Syst. I* **70** 2037–47
[2] Abbasian E and Gholipour M 2022 Design of a highly stable and robust 10T SRAM cell for low-power portable applications *Circuits Syst. Signal Process.* **41** 5914–32
[3] Kumar S and Mukherjee A 2023 Design of soft-error resilient SRAM cell with high read and write stability for robust operations *AEU-International Journal of Electronics and Communications* **168** 154719
[4] Xue X, Sai Kumar A, Khalaf O I, Somineni R P, Abdulsahib G M, Sujith A, Dhanuja T and Vinay M V 2023 Design and performance analysis of 32 × 32 memory array SRAM for low-power applications *Electronics* **12** 834
[5] SandeepRaval 2014 System Verilog-based Verification of Write Operation in SDRAM using Memory Controller *International Journal for Scientific Research & Development* **2** 2014 242–4
[6] Vutukuri V, Adusumilli V B, Uppu P K, Varsa S and Thummala R K 2020 Verification of SDRAM controller using systemverilog *2020 IEEE Int. Conf. on Electronics, Computing and Communication Technologies (CONECCT)* (IEEE) pp 1–6
[7] Mambu K, Charles H P, Kooli M and Dumas J 2022 Towards integration of a dedicated memory controller and its instruction set to improve performance of systems containing computational SRAM *Journal of Low Power Electronics and Applications* **12** 18
[8] Juracy L R, de Morais Amory A and Moraes F G 2022 A fast, accurate, and comprehensive PPA estimation of convolutional hardware accelerators *IEEE Transactions on Circuits and Systems I: Regular Papers* **69** 5171–84
[9] Praveen K N and Shivaleelavathi B G 2015 SRAM Memory Layout Design in 180nm Technology *IJERT-International Journal of Engineering Research and Technology* **4** 2015
[10] Rawat K, Sahni K and Pandey S 2014 Design of AMBA APB bridge with reset controller for efficient power consumption *2014 9th Int. Conf. on Industrial and Information Systems (ICIIS)* (IEEE) pp 1–5
[11] Prathibha G and AmbikaSekhar 2013 APB bridge based on AMBA 4.0 *International Journal Of Engineering Research & Technology (IJERT)* **02** 09

[12] Girdhar D and Shukla N K 2014 Design and verification of AMBA APB protocol *International Journal of Computer Applications* **95** 21
[13] Manu B N and Prabhavathi P 2013 Design and implementation of AMBA ASB APB Bridge *2013 Int. Conf. on Fuzzy Theory and Its Applications (iFUZZY)* (IEEE) pp 234–8
[14] Divekar S and Tiwari A 2014 Multichannel AMBA AHB with multiple arbitration techniques *2014 Int. Conf. on Communication and Signal Processing* (IEEE) pp 1854–8
[15] Kumar M K, Sajja A and Noorbasha F 2017 Design and FPGA Implementation of AMBA APB bridge with clock skew minimization technique *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)* **7** 42–5
[16] Mukunthan J, Raj A D, Keerthivadana S, Kiruthika R, Shruthi T and Snegasri S 2021 design and implementation of AMBA APB protocol *IOP Conf. Ser.: Mater. Sci. Eng.* (IOP Publishing) vol 1084, 012050
[17] Pal S, Sri D D, Ki W H and Islam A 2020 Highly stable low-power radiation-hardened memory-by-design SRAM for space applications *IEEE Trans. Circuits Syst. Express Briefs* **68** 2147–51
[18] Ma C, Liu Z and Ma X 2011 Design and implementation of APB bridge based on AMBA 4.0 *2011 Int. Conf. on Consumer Electronics, Communications and Networks (CECNet)* (IEEE) pp 193–6
[19] Rawat K, Sahni K and Pandey S 2015 RTL Implementation for AMBA ASB APB Protocol at System on Chip level *2015 2nd Int. Conf. on Signal Processing and Integrated Networks (SPIN)* (IEEE) pp 927–30
[20] Goswami D and Shah D J 2015 Implementation of system verilog environment for functional verification of AHB-DMA bridge *International Journal on Recent and Innovation Trends in Computing and Communication* **3** 2799–802
[21] Heli Shah P, Chinmaymodi P and BhargavTarpara P 2015 Design & implementation of advance peripheral bus protocol *International Journal of scientific engineering and applied science (IJSEAS)* **1** 2015
[22] Kumar A and Sinha R 2011 Design and Verification analysis of APB3 Protocol with Coverage *International Journal of Advances in Engineering & Technology* **1** 310
[23] Sanwatsarkar M and Nagar J 2015 Implementation of DDR SDRAM controller using verilog HDL *IOSR Journal of Electronics and Communication Engineering* **10** 69–74
[24] Fischer L and Pyatnychko Y 2012 FPGA design for DDR3 memory. WPI Worcester. https://digital.wpi.edu/concern/student_works/2514nn20r
[25] Solokhina T, Petrichkovich J, Glushkov A, Belyaev A, Menshenin L, Putrya F, Yuriy S and Elena S 2016 Radiation tolerant heterogeneous multicore 'system on chip' with built-in multichannel spacefibre switch for onboard data management and mass storage device: components, short paper *2016 Int. SpaceWire Conf. (SpaceWire)* (IEEE) pp 1–6
[26] Bibay P, Kumar Sahu A and Kumar Chandra V 2013 Design and implementation of DDR SDRAM controller using verilog *International Journal of Science and Research (IJSR)* **2** 320–4
[27] MP P K and Panda S K 2019 Design and verification of DDR SDRAM memory controller using systemverilog for higher coverage *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (IEEE) pp 689–94
[28] Tamilselvi M, Vedhanayagi P and Ramasamy K 2017 VLSI design of low power fault detection in SRAM using BIST *International journal of advanced research in electrical, electronics and instrumentation engineering* **6** 1–9
[29] Reddy K S, Soujanya P and Sudha D K 2020 ASIC design and verification of AMBA APB protocol using UVM *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* **9** 636–42
[30] Hiremath S and Vinayakgouda Y 2021 RTL to GDSII implementation of Advanced High-Performance Bus Lite *2021 6th Int. Conf. on Communication and Electronics Systems (ICCES)* (IEEE) pp 303–6
[31] Calhoun B H and Chandrakasan A P 2007 A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation *IEEE J. Solid-State Circuits* **42** 680–8
[32] Ebrahimi B, Afzali-Kusha H and Afzali-Kusha A 2012 Low power and robust 8T/10T subthreshold SRAM cells *2012 Int. Conf. on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)* (IEEE) pp 141–4
[33] Pal S and Islam A 2015 Variation tolerant differential 8T SRAM cell for ultra-low power applications *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **35** 549–58
[34] Gupta S, Gupta K and Pandey N 2017 A 32-nm subthreshold 7T SRAM bit cell with read assist *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **25** 3473–83
[35] Patel P K, Malik M M and Gupta T K 2018 Reliable high-yield CNTFET-based 9T SRAM operating near threshold voltage region *J. Comput. Electron.* **17** 774–83
[36] Sachdeva A and Tomar V K 2021 Stability and dynamic power analysis of novel 9T SRAM cell for IoT applications *In Communication and Intelligent Systems: Proceedings of ICCIS 2020* (Springer Singapore) pp 945–53
[37] Jaiswal A, Agrawal A, Ali M F, Sharmin S and Roy K 2020 I-SRAM: interleaved wordlines for vector boolean operations using SRAMs *IEEE Trans. Circuits Syst. I* **67** 4651–9
[38] Li X, Qin J, Huang B, Zhang X and Bernstein J B 2006 SRAM circuit-failure modeling and reliability simulation with SPICE *IEEE Trans. Device Mater. Reliab.* **6** 235–46
[39] Yang X, Wu N and Andrian J H 2016 A novel bus transfer mode (AS transfer) and a performance evaluation methodology *Integration* **52** 23–33
[40] Yang X and Andrian J H 2014 A high-performance on-chip bus (MSBUS) design and verification *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **23** 1350–4
[41] Narendrappa S, RadhaSwamy S and He L 2018 Performance verification of multi-master AHB bus system *Proc. of the World Congress on Engineering* 1, 2018
[42] Jain P and Rao S 2021 Design and verification of advanced microcontroller bus architecture-advanced peripheral bus (AMBA-APB) protocol *2021 Third Int. Conf. on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (IEEE) pp 462–7
[43] Krishnegowda D 2021 Developing a bus functional model for APB slave using universal verification methodology *2021 s Int. Conf. on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)* (IEEE) pp 1–5
[44] Sahu N, Kumar A and Kandari R 2022 Design and verification of APB IP core using different verification methodologies *2022 International Conference on Breakthrough in Heuristics And Reciprocation of Advanced Technologies (BHARAT)* (IEEE) pp 150–4