Application Specific Power Optimization Methodology for IOT Edge Nodes

Preetham Gowda M N[1], Prof. Pushpanjali J[2], *

[1] PG student, Dept. of ECE, Bangalore institute of technology, and Bengaluru | Email: preethamgowdamn03@gmail.com
[2] Associate Professor Dept. of ECE, Bangalore institute of technology, and Bengaluru | Email: jpushpanjali66@gmail.com

Abstract — The need for ultra-low-power embedded processors that can function well under severe energy limitations is rising due to the Internet of Things' (IoT) edge computing systems' quick development. Due to constant switching activity during idle computing intervals, conventional microcontroller designs frequently incur high dynamic power dissipation, which lowers overall energy efficiency and operational lifespan. This paper uses a clock-gated openMSP430 processor architecture to propose an application-specific power optimization solution for Internet of Things edge nodes. The suggested method suppresses redundant switching transitions during idle execution periods by integrating fine-grained clock gating methods into certain sequential portions of the processor data path and control logic. In order to assess power, timing, and area characteristics, the architecture is generated using power-aware synthesis techniques and implemented at Register Transfer Level (RTL) using Verilog HDL. The optimization approach concentrates on reducing dynamic power consumption without compromising temporal stability, instruction execution precision, or processor functionality. The results of experimental synthesis show a significant decrease in switching power and total energy consumption while preserving architectural dependability appropriate for embedded Internet of Things applications. For battery-operated edge devices, wireless sensor nodes, and portable low-power embedded systems needing longer operating lifetimes and increased energy efficiency, the suggested architecture is ideal.

Keywords — IoT Edge Nodes; openMSP430; Clock Gating; Low-Power VLSI; Power-Aware Synthesis.

I. INTRODUCTION

The development of Internet of Things (IoT) technology has led to the widespread use of edge-processing and intelligent sensing devices in consumer electronics, smart agriculture, healthcare monitoring, environmental sensing, industrial automation, and autonomous mobility. IoT edge nodes, in contrast to conventional centralized computing infrastructures, are made to process data locally close to the source of creation, which lowers network congestion, improves real-time responsiveness, and minimizes communication delay. Nonetheless, most edge devices run on limited energy sources such small batteries, energy harvesters, or low-power portable supplies. As a result, one of the most important design goals in embedded system development is now power efficiency.

Because of constant clock propagation and switching transitions inside sequential logic parts, the processor core is one of the many components of embedded systems that significantly contributes to total dynamic power consumption. Regardless of actual computing activity, clock networks in synchronous digital systems are always in operation. Even though they are not involved in data processing, many processor modules continue to receive clock transitions during idle or partially active execution cycles. Particularly in CPUs meant for sporadic IoT applications, these repeated switching transitions lead to needless dynamic power usage.

Reducing dynamic power consumption inside the CPU without compromising computational stability or instruction execution behaviour is the main goal of this effort. The suggested architecture is designed especially for IoT edge workloads that exhibit event-driven execution patterns, sporadic computing activity, and periodic sensing activities. The architecture reduces needless switching transitions and increases energy efficiency appropriate for battery-operated computers by taking advantage of idle execution periods.

Developing a synthesis-oriented low-power methodology for embedded processor architectures, implementing selective clock gating within the openMSP430 core, analyzing dynamic power reduction in detail, and assessing timing and area trade-offs related to clock-aware optimization strategies are all examples of the research contribution of this work. The results show how application-specific clock gating may effectively reduce CPU power consumption while preserving stable system operation, which is appropriate for contemporary low-power Internet of Things applications.

II. ANALYSIS OF OPENMSP430 ARCHITECTURE

A small 16-bit RISC processor made for embedded low-power applications is the openMSP430 processor architecture. The CPU has a modular design made up of many linked functional blocks that handle interrupts, arithmetic operations, memory interfacing, control-flow management, and instruction execution.

The Arithmetic Logic Unit (ALU), instruction decoder, register file, program counter, memory controller, interrupt controller, and execution control logic are among the main architectural elements of the CPU. When compared to high-performance CPU designs, these modules collectively enable effective instruction processing while preserving comparatively minimal hardware complexity.

Because clock signals continually flow across the processor regardless of active computing requirements, the traditional openMSP430 architecture nonetheless suffers substantial switching activity among sequential components despite its low-power design approach. Even when internal state values stay the same, inactive modules continue to receive clock transitions during idle execution phases. Dynamic power dissipation is directly impacted by this pointless switching.

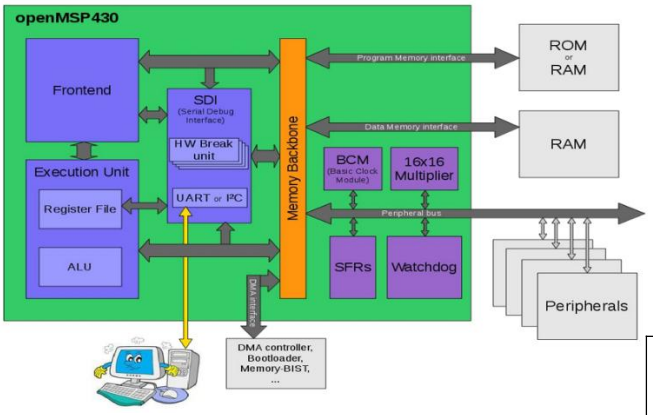

Fig. 1 openMSP430 architecture

III. BACKGROUND AND RELATED WORK

Energy-sensitive applications and aggressive transistor growth have made low-power design techniques more crucial in contemporary VLSI systems. The charging and discharging of capacitive nodes during logic transitions is the main cause of dynamic power dissipation, which accounts for the majority of overall power consumption in synchronous digital circuits.

In CMOS circuits, the dynamic power used is shown as:

$$P_{dynamic} = \alpha C V_{DD}^2 f \quad (1)$$

Where:

α = switching activity factor

C = load capacitance

VDD = supply voltage

f = operating frequency

Switching activity is one of these factors that greatly leads to needless power waste in embedded CPUs. Because clock signals constantly toggle regardless of computing activity, clock networks account for a sizable portion of all switching transitions.

By selectively preventing clock propagation to idle modules, clock gating reduces switching activity. Clock transitions are prevented when enable conditions are dormant, which lowers consecutive toggling and related dynamic power.

IV. PROPOSED METHODOLOGY

The proposed method focuses on using fine-grained clock gating techniques appropriate for Internet of Things edge-node applications to reduce dynamic power consumption in the openMSP430 processor architecture. Clock signals continually flow through all sequential components of traditional synchronous embedded computers, regardless of actual computational activity. Inactive modules thus experience needless switching transitions, which raise dynamic power consumption. Reducing redundant switching activity is crucial for increasing energy efficiency and prolonging operational lifespan since IoT edge devices often operate under limited energy settings and display intermittent workloads.

TABLE I
Functional Description of Proposed Clock-Gated openMSP430 Architecture

Module	**Function**	**Power Optimization Objective**
Clock Gating Controller	creates gated clock signals according to activity levels	minimizes needless clock switching activities
Enable Generation Logic	uses processor execution states to generate enable signals	regulates the spread of selective clocks
Datapath Registers	keeps data from intermediate computations	reduces the amount of unnecessary consecutive toggling
Control Unit	regulates the flow of	Inactive control-state

	instructions	transitions are suppressed
Memory Interface	manages data transmission and instruction	minimizes switching between idle memory access
openMSP430 Core	carries out embedded processing tasks	keeps the CPU operating at low power

A. Fine-Grained Clock Gating Technique

In order to minimize unnecessary switching activity inside dormant sequential modules, clock gating is used. The suggested approach avoids timing instability and synchronization problems by using fine-grained clock gating rather than broad clock suppression. Clock signals are only conditionally transmitted in this method when module activity is observed. In order to avoid needless transitions inside flip-flops and sequential logic structures, a module's related clock signal is suppressed when it is idle during a certain execution cycle.

The operation of the gated clock is shown as:

$$CLK_{gated} = CLK \cdot EN \qquad (2)$$

where EN stands for the enable signal produced based on processor activity circumstances and CLK for the original clock signal. Register update needs, operational dependence circumstances, and instruction execution states are used to dynamically create the enable logic. Only active modules get clock transitions thanks to this selective gating system, which keeps idle modules away from switching activity.

Figure 2 shows the fine-grained clock gating structure implemented in the proposed architecture. The figure should include the original clock signal, enable logic, gated clock generation block, and sequential elements controlled through activity-based gating conditions.

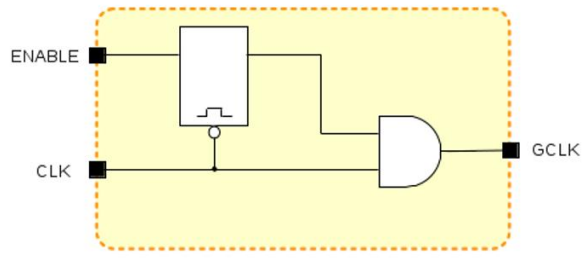

Fig. 2: OpenMSP430's Fine-Grained Clock Gating Structure

B. Enable Signal Generation Based on Activity

The precise creation of enable signals is essential to clock gating's effectiveness. According to processor operating activity, enable signals are created in the suggested manner. In order to ascertain if clock propagation is required for a certain module, the enable-generation logic continually observes processor execution states. The associated modules stay active and continue to receive clock transitions during activities such as register update, memory access, arithmetic execution, instruction fetch, and decoding. However, a module's clock signal is deactivated to prevent needless switching activity when it is not involved in the current execution cycle.

Switching capacitance related to consecutive elements is minimized and unnecessary toggling inside idle datapath structures is greatly reduced by the activity-monitoring method. The CPU often goes through periods of dormant operation because IoT edge-node applications require periodic sensing and event-driven processing. By taking advantage of these idle times, the suggested enable-generation technique efficiently increases energy efficiency while preserving steady processor performance.

C. Synthesis Flow with Power Awareness

Power-aware synthesis techniques are used to create the suggested clock-gated design, which is implemented using Verilog HDL. RTL elaboration, logic optimization, clock optimization, technology mapping, and power analysis are all included in the synthesis pipeline. In order to maintain clock-gating structures during optimization phases and guarantee correct implementation of gated sequential logic, power-aware synthesis directives are enabled.

Power reports, time analysis reports, switching activity estimation, and area usage reports under various operating situations are used to assess the synthesized architecture. The main objective of the evaluation is to reduce dynamic power by minimizing clock activity in modules that are not in use. To confirm that the added gating logic does not considerably increase area overhead or impair timing performance, more study is carried out.

V. RESULTS AND DISCUSSION

A. Baseline openMSP430 Architecture

The synthesis report of the openMSP430 provides detailed insights into the resource utilization, timing analysis, and overall performance metrics, ensuring that the design meets the specified requirements for functionality and efficiency.

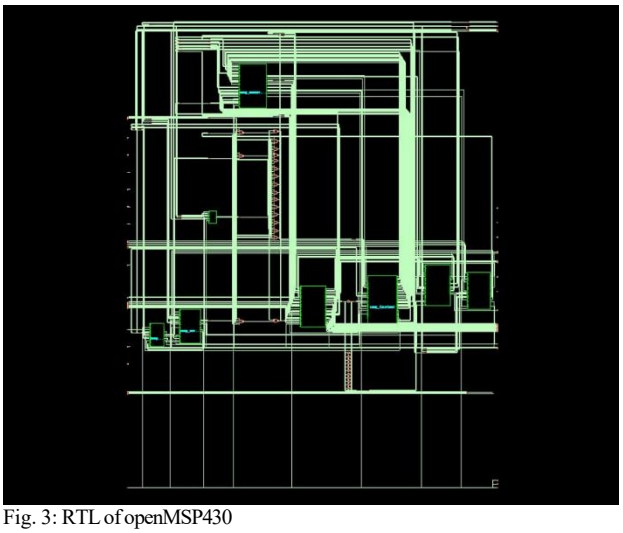

Fig. 3: RTL of openMSP430

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           May 14 2026  01:14:06 pm
  Module:                 openMSP430
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

       Instance              Module        Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------------------------
openMSP430                                      5018  38383.156     0.000    38383.156 <none> (D)
  clock_module_0          omsp_clock_module       97    996.080     0.000      996.080 <none> (D)
    sync_cell_lfxt_clk    omsp_sync_cell           3     43.143     0.000       43.143 <none> (D)
    sync_cell_puc         omsp_sync_cell_1478      3     43.143     0.000       43.143 <none> (D)
    sync_reset_por        omsp_sync_reset          3     56.767     0.000       56.767 <none> (D)
  dbg_0                   omsp_dbg               764   6006.001     0.000     6006.001 <none> (D)
    dbg_uart_0            omsp_dbg_uart          348   2894.386     0.000     2894.386 <none> (D)
      sync_cell_uart_rxd  omsp_sync_cell_1476      3     43.143     0.000       43.143 <none> (D)
  execution_unit_0        omsp_execution_unit   2015  15353.716     0.000    15353.716 <none> (D)
    alu_0                 omsp_alu               699   3362.907     0.000     3362.907 <none> (D)
    register_file_0       omsp_register_file    1003   9823.048     0.000     9823.048 <none> (D)
  frontend_0              omsp_frontend          719   5288.460     0.000     5288.460 <none> (D)
  mem_backbone_0          omsp_mem_backbone      309   2390.290     0.000     2390.290 <none> (D)
  multiplier_0            omsp_multiplier        916   6891.574     0.000     6891.574 <none> (D)
  sfr_0                   omsp_sfr                41    270.213     0.000      270.213 <none> (D)
    sync_cell_nmi         omsp_sync_cell_1477      3     43.143     0.000       43.143 <none> (D)
  watchdog_0              omsp_watchdog          126   1013.489     0.000     1013.489 <none> (D)
  (D) = wireload is default in technology library
```

Fig. 4 Area Report of openMSP430 without Clock Gating

```
Instance: /openMSP430
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------------
    Category         Leakage      Internal     Switching        Total     Row%
  ------------------------------------------------------------------------------
       memory     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
     register     8.77364e-05   1.91880e-03   6.63611e-05   2.07290e-03  59.14%
        latch     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
        logic     9.09496e-05   7.85819e-04   4.34892e-04   1.31166e-03  37.42%
         bbox     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
        clock     0.00000e+00   0.00000e+00   1.20234e-04   1.20234e-04   3.43%
          pad     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
           pm     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
  ------------------------------------------------------------------------------
    Subtotal     1.78686e-04   2.70462e-03   6.21487e-04   3.50480e-03  99.99%
  Percentage          5.10%        77.17%        17.73%       100.00% 100.00%
  ------------------------------------------------------------------------------
```

Fig. 5: Power Report of openMSP430 without Clock Gating

TABLE II
POWER AND AREA OF OPENMSP430 WITHOUT CLOCK GATING

Area Consumed (µm²)	Power Consumed (mw)
38383.156	3.50480

The area consumed by the design is 38383.156µm², with a power consumption of 3.50480mW, indicating a compact and energy-efficient implementation.

B. CLOCK-GATED OPENMSP430 ARCHITECTURE

The revised openMSP430 design reduces dynamic power consumption by reducing needless clock switching activity while retaining acceptable timing performance, as shown by the synthesis area and power report with clock gating.

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           May 14 2026  02:14:06 pm
  Module:                 openMSP430
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

       Instance              Module        Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------------------------
openMSP430                                      5516  37169.845     0.000    37169.845 <none> (D)
  clock_module_0          omsp_clock_module       99    999.865     0.000      999.865 <none> (D)
    sync_cell_lfxt_clk    omsp_sync_cell           3     43.143     0.000       43.143 <none> (D)
    sync_cell_puc         omsp_sync_cell_1478      3     43.143     0.000       43.143 <none> (D)
    sync_reset_por        omsp_sync_reset          3     56.767     0.000       56.767 <none> (D)
  dbg_0                   omsp_dbg               876   6021.139     0.000     6021.139 <none> (D)
    dbg_uart_0            omsp_dbg_uart          363   2809.613     0.000     2809.613 <none> (D)
      sync_cell_uart_rxd  omsp_sync_cell_1476      3     43.143     0.000       43.143 <none> (D)
  execution_unit_0        omsp_execution_unit   2179  14692.186     0.000    14692.186 <none> (D)
    alu_0                 omsp_alu               824   3446.166     0.000     3446.166 <none> (D)
    register_file_0       omsp_register_file    1020   9142.595     0.000     9142.595 <none> (D)
  frontend_0              omsp_frontend          751   4999.325     0.000     4999.325 <none> (D)
  mem_backbone_0          omsp_mem_backbone      395   2381.964     0.000     2381.964 <none> (D)
  multiplier_0            omsp_multiplier       1015   6736.410     0.000     6736.410 <none> (D)
  sfr_0                   omsp_sfr                40    266.429     0.000      266.429 <none> (D)
    sync_cell_nmi         omsp_sync_cell_1477      3     43.143     0.000       43.143 <none> (D)
  watchdog_0              omsp_watchdog          130    931.744     0.000      931.744 <none> (D)
  (D) = wireload is default in technology library


Instance: /openMSP430
Power Unit: W
PDB Frames: /stim#0/frame#0
  ------------------------------------------------------------------------------
    Category         Leakage      Internal     Switching        Total     Row%
  ------------------------------------------------------------------------------
       memory     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
     register     7.21411e-05   6.72420e-04   2.74185e-05   7.71980e-04  47.45%
        latch     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
        logic     6.23853e-05   4.17148e-04   2.52299e-04   7.31832e-04  44.98%
         bbox     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
        clock     4.78754e-06   7.60812e-05   4.21906e-05   1.23059e-04   7.56%
          pad     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
           pm     0.00000e+00   0.00000e+00   0.00000e+00   0.00000e+00   0.00%
  ------------------------------------------------------------------------------
    Subtotal     1.39314e-04   1.16565e-03   3.21908e-04   1.62687e-03  99.99%
  Percentage          8.56%        71.65%        19.79%       100.00% 100.00%
  ------------------------------------------------------------------------------
```

Fig. 5: Power and Area Report of openMSP430 with clock gating

TABLE III
POWER AND AREA OF OPENMSP430 WITH CLOCK GATING

Area Consumed (µm²)	Power Consumed (mw)
37169.845	1.62687

C. COMPARATIVE ANALYSIS

TABLE IV
POWER AND AREA OF OPENMSP430 COMPARISION

Parameter Analyzed	Clock Gated openMSP 430	NON-Clock Gated openMSP 430	Improvement
Power Consumed (mw)	1.62687	3.50480	3.2%
Area Consumed (µm²)	37169.845	38383.156	53.6%

The analysis reveals that the clock gated openMSP430 consumes 1.62687mW of power and occupies 37169.845µm², significantly outperforming the non-clock gated openMSP430, which consumes 3.50480mW and occupies 38383.156 µm², highlighting the efficiency of the clock gating technique. The simultaneous reduction in power consumption and area in the clock gated openMSP430 is achieved through the implementation of clock gating techniques that disable the clock signal to inactive modules, thereby minimizing switching activity and energy usage, while also allowing for a more compact design by eliminating unnecessary circuitry associated with constant clocking.

VI. CONCLUSION AND FUTURE WORK

The openMSP430 processor architecture's RTL-level low-power optimization was effectively accomplished through the use of clock gating, operand isolation, conditional register updates, and switching activity reduction approaches. The clock network, ALU, register file, memory interface, and decoder logic were found to be the primary sources of dynamic power consumption in a thorough baseline RTL study. While maintaining functional correctness, time stability, and synthesizability, the suggested optimization approach successfully decreased needless switching activity across combinational and sequential logic. Following optimization, proper processor functioning was verified using functional verification and synthesis analysis. With just 3.2% area reduction following synthesis, the improved architecture reduced dynamic power usage by about 53% as compared to the baseline design. The collected findings show that optimizing embedded processor designs for FPGA and ASIC implementations at the RTL level is an efficient and technology-independent way to increase energy efficiency.

VII. ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Chandra Mohan Umapathy and Prof. Pushpanjali J. for their invaluable guidance, constant encouragement, and technical support throughout the course of this research work. Their insightful suggestions, constructive feedback, and expertise greatly helped me in understanding the technical aspects of the project and overcoming various challenges encountered during its execution.

I am deeply thankful for their patience, motivation, and continuous mentorship, which played a significant role in the successful completion of this work. Their support, encouragement, and willingness to share their knowledge have been instrumental in shaping both this research and my academic growth. I sincerely appreciate their guidance and contributions, which have greatly enriched my learning experience and helped me successfully accomplish this work..

VIII. REFERENCES

[1] S. Catrinoiu and M. Enachescu "openMSP430-based Dual-Core Lockstep ASIC Design for ASIL D Applications" 2025 International Semiconductor Conference (CAS) | 2025 | Bucharest, Romania.

[2] N. Jyothsna and T. Lalith Kumar "Physical Design Implementation of OpenMSP430 Using Different Approaches" SSRG International Journal of Electronics and Communication Engineering | May 2020 | Kadapa, India.

[3] S. K. Varunkumar, B. J. Pallavi, G. N. Tushar, V. Prathiksha, and Ramayshree "Digital IC Characterization and Testing Using MSP430 Microcontroller" ICRTT Conference Proceedings, International Journal of Engineering Research & Technology (IJERT) | 2018 | Ujire, India.

[4] O. Girard "openMSP430" OpenCores | 6 November 2017 | www.opencores.org.

[5] S. Sondon, P. Mandolesi, F. Masson, and P. Julián "A Dual Core Low Power Microcontroller with openMSP430 Architecture for High Reliability Lockstep Applications Using a 180 nm High Voltage Technology Node" 2013 IEEE 4th Latin American Symposium on Circuits and Systems | 2013 | Argentina

[6] Bansal, V., Mohamed, O. A., & Ghaffari, F. (2023). Layout-based reliability analysis of openMSP430 register file under external radiations. 2023 International Conference on Microelectronics (ICM), 294–297. https://doi.org/10.1109/icm60448.2023.10378890.