Application Note

AM62A Edge AI Retail Scanner Demo: Analysis for SoC Selection and Power Usage

Reese Grimsley and Colin Callaghan

ABSTRACT

TI’s AM6xA Processors are designed for vision applications requiring intensive image analytics. Retail checkout and scanner applications like item recognition, barcode scanning and decoding, and theft detection benefit from imaging and vision analytics to improve accuracy, speed, and generality to new environments. This application note analyzes a retail checkout demo application, which uses a raw camera sensor and runs a gstreamer-based application with deep learning for object detection, on the AM62A's heterogeneous architecture. The core load across the AM62A processor is used to select a cost-optimized version of the system-on-chip (SoC) and the application's power usage shows SoC consumpion is under 2 Watts active power.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

Copyright © 2023 Texas Instruments Incorporated

1 Introduction

Smart camera applications in retail sectors are increasing in popularity given the rich information content in imagery as well as the improving capability of processing that data at the edge. Use-cases like checkout scanners, barcode images, asset and people tracking, theft detection, and more are helping to automate, simplify, and streamline customers' experiences.

Although vision is an intuitive concept for humans, computer vision is challenging. Imagery is information-dense and visual patterns can occur in many shapes, sizes, and contexts. Conventional computer vision using filters, transforms, and specialized algorithms are effective; However, there algorithms are often difficult to accelerate in their entirety and can require careful tuning to the environment and specifics of the task. Machine learning and deep learning approaches to image processing are much more generalized and often boast higher accuracy. While these learning-based algorithms (AI) are often more computationally complex, they are also easier to accelerate given their strong reliance on matrix math.

Processing imagery with deep learning and AI in the cloud be done easily, yet comes at more significant and recurring cost than processing locally. Reactive applications like checkouts would be slow and frustrating for customers due to network latency. Security applications can cause privacy concerns. Furthermore, as the solutions scale, the associated cloud costs will scale similarly. Processing imagery locally on the device that captures video solves these issues, but requires a processor that matches requirements for cost, power, and performance.

2 AM62A Processor

The AM62A Edge AI Microprocessor, shown in Figure 2-1, is designed for single or dual camera applications that are cost and power sensitive, yet require intensive image analysis. Vision applications benefit from hardware acceleration to enhance image quality, speed up preprocessing, and accelerate analytics algorithms like deep neural networks.

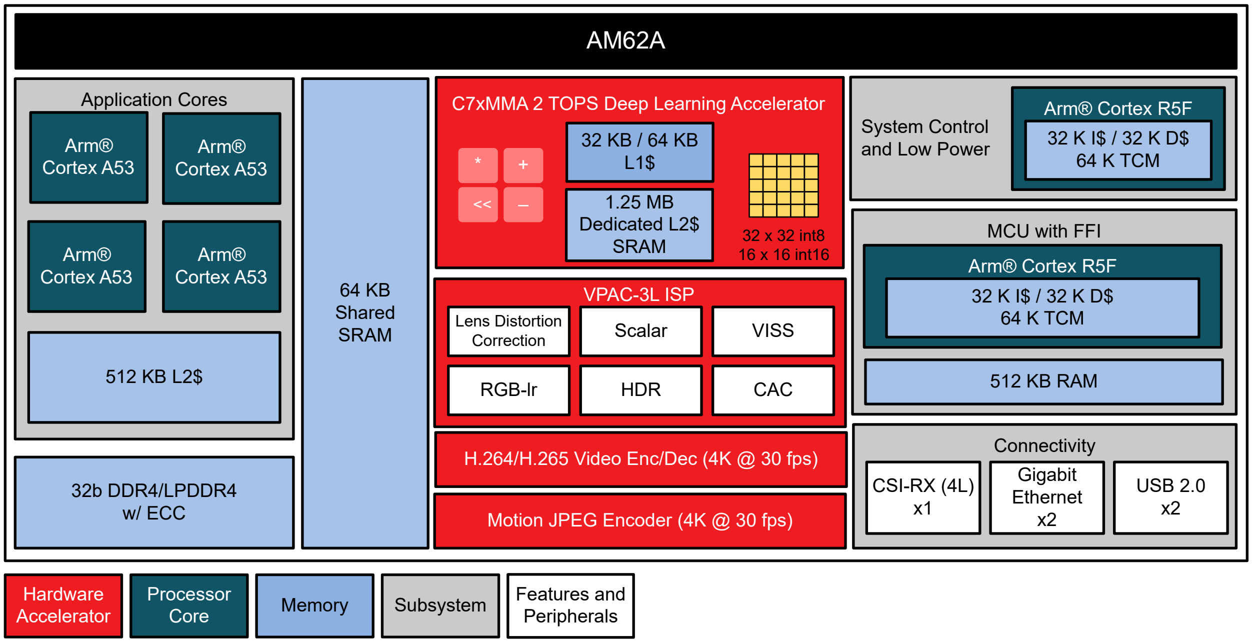


Figure 2-1. AM62A Simplified Block Diagram

Copyright © 2023 Texas Instruments Incorporated

Figure 2-2 demonstrates a general dataflow for vision analytics applications. Images are produced by a low-cost raw image sensor, that is, the camera. Raw image data enters the processor through the 4-lane MIPI-CSI2 port, which can be split into multiple virtual channels for more cameras. The image is enhanced by the ISP to reduce noise, tune white balancing and gain, filter and interpolate color information, and process High Dynamic Range (HDR) information. For applications with wide angle lens, the Lens Distortion Correction (LDC) accelerator reduces warping effects from the lens. After preprocessing the image to meet the AI model's input specification, the hardware accelerator runs the model at 50-100x the CPU's capability; see Table 3-1 for several model benchmarks. An AI model can accomplish tasks like recognize food items, locate a barcode on a package, identify where customers spend the most time, or detect patterns of theft.

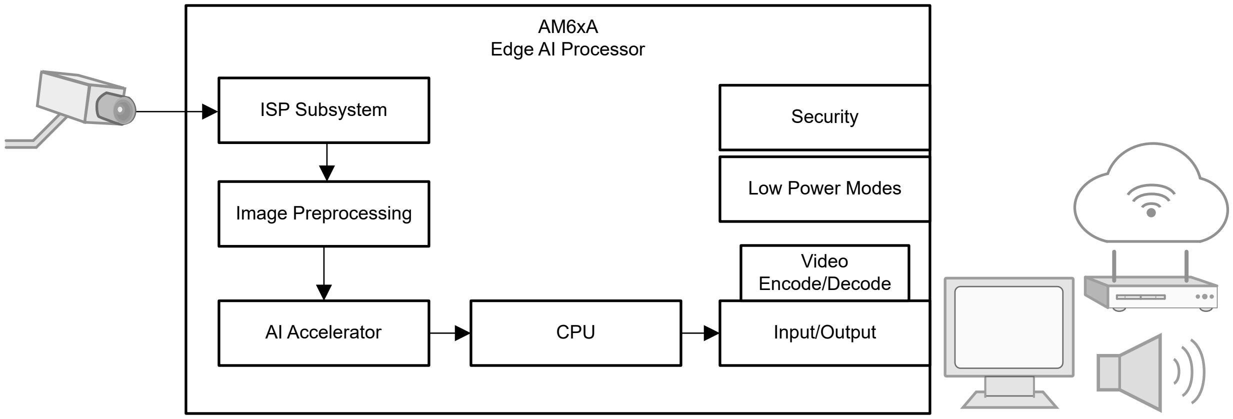


Figure 2-2. AM6xA Vision Application Data Flow

Once the AI model has run, the specific application can decide how to act upon the information such as communicating over the network, showing information on a display, or playing an alarm sound. When inactive, low power modes drastically reduce power consumption; when running at 100% load, the SoC consumes less than 3 Watts at up to 85°C, reducing the need for active cooling. On-device security prevents tampering to protect data and firmware IP.

The AM62A features a 2 TOPS deep learning accelerator that is designed in-house. The accelerator is composed of a 256-bit C7x DSP tightly coupled to a matrix multiply accelerator (MMA). This tight coupling enables fast and efficient data movement to the accelerator, which ensures high utilization of the accelerator. The 2 TOPS metric refers the max number of operations per second on 8-bit quantized matrices. However, TOPS is not an ideal indicator of performance for deep learning acceleration, because 1 TOPS can have very different inference time and power usage based on the accelerator architecture and even model / neural network architecture. For this reason, it is more informative to view benchmarks that show inference rate (frames per second).

3 Deep Learning Benchmarks

Table 3-1 shows benchmarks on the AM62A running the Linux Edge AI SDK version 8.6 on the Starter Kit EVM revision E2. Note that these numbers reflect 1.7 TOPS of maximum performance, as the E2 EVM's PMIC supplies 0.75 VDD core voltage. Achieving the full 2 TOPS requires 0.85 VDD. EVMs after E2 will use the updated PMIC for full performance entitlement.

Table 3-1. Benchmarks on the AM62A

Model Name	Accuracy (C7xMMA)	Frames-per-Second (C7xMMA)	Frames-per-Second (CPU)	Resolution
Classification Network – Accuracy is Top1 Metric				
TFL-CL-0010-mobileNetV2	74.51	251	5.5	224 x 224
Object Detection Network – Accuracy is mAP50-95				
TFL-OD-2000-ssd-mobV1-coco-mlperf-300x300	26.14	152	2.4	300 x 300

Copyright © 2023 Texas Instruments Incorporated

TI's model zoo hosts many more models that are pretrained and fully accelerated on the C7xMMA. The most up-to-date benchmarks (relative to software and EVM version) can be found in TI's Model Analyzer tool [1].

4 Retail Checkout Scanner Application

A reference application was developed for the AM62A to showcase its capabilities for an automated checkout system using an object detection neural network. A custom model was trained to recognize a dozen different food items, and a Linux Python3 application was written around this model using TI's gstreamer plugins to leverage hardware acceleration where possible. Application source code [2] and in-depth description on how the demo works are available on Github. For further guidance on building an application like this, see the associated application note [3]. The block diagram in Figure 4-1 depicts the application flow within gstreamer and how various plugins execute on remote cores within the SoC.

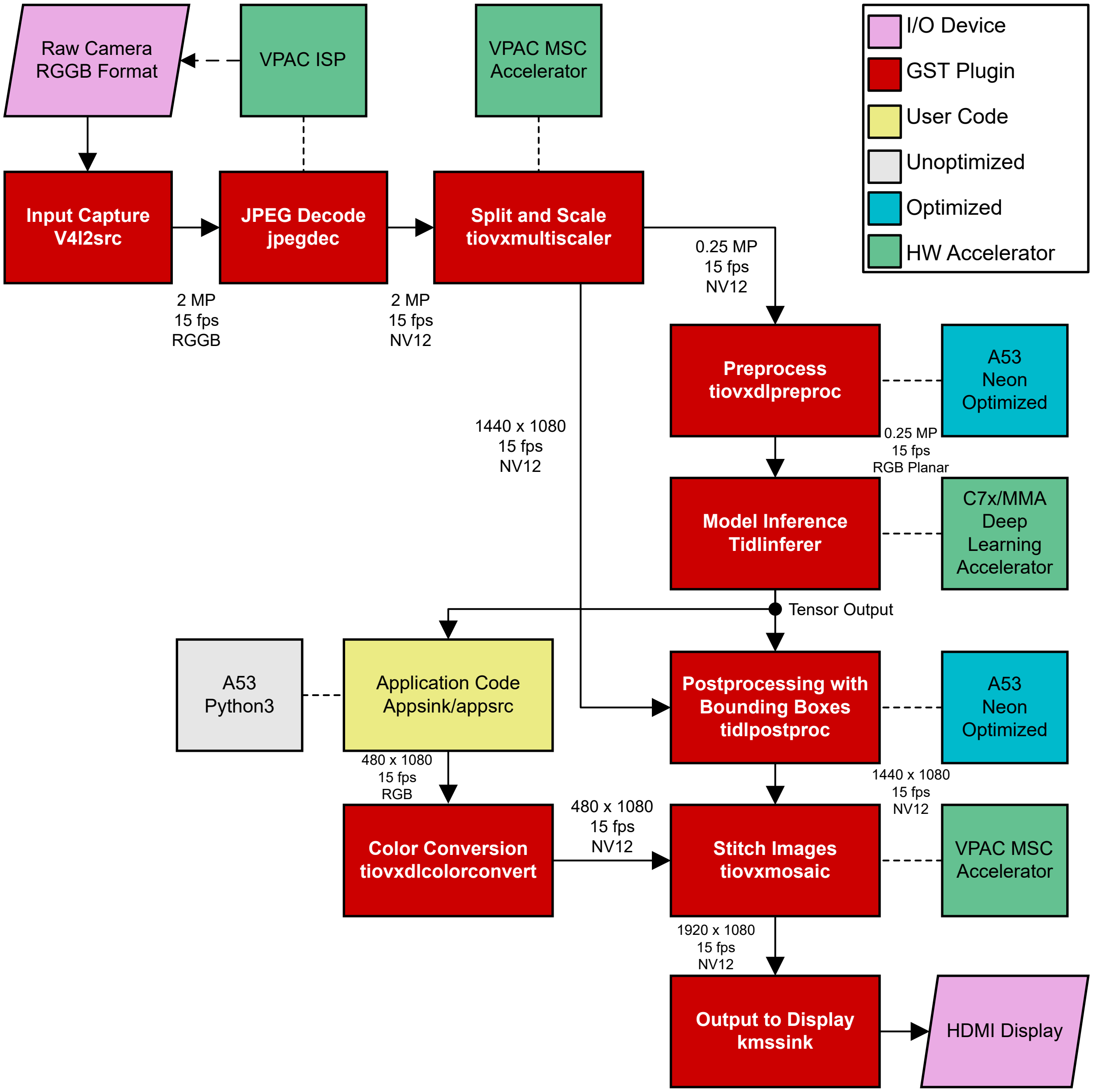


Figure 4-1. Retail Checkout Application Flow With Resolutions and Pixel Formats. (30fps is the maximum achievable; the application is bottlenecked by the application code such that FPS is closer to 15)

This document analyzes this application and uses the core load both to guide selection of a suitable AM62A variant as well as provide a power usage estimate using the Power Estimation Tool [4]. This analysis can be followed for other applications designed and benchmarked on the Starter Kit EVM, which uses the superset variant AM62A74 (2 TOPS acceleration, 4 Arm® Cortex A53 cores).

Copyright © 2023 Texas Instruments Incorporated

5 Core Loading

Processors in the AM6xA family have heterogeneous architectures with a variety of Arm® cores and hardware accelerators. Typical load tools, like 'top' in Linux, do not show the load on integrated microcontrollers or hardware accelerators.

The application under test is the Retail Checkout demo, which runs at around 15 fps with approximately 200 ms of latency per frame. This is bottlenecked primarily by application code written in Python3 for the CPU. The camera and object detection model, mobilenetv2SSD, can handle much higher framerate (up to 60). For a checkout scanner, 15 fps would be sufficient and will allow the developer to select a cost-down variant of the SoC.

In the 8.6 SDK, two methods for viewing core loads are supported:

- The tiperfoverlay gstreamer plugin, which draws bars along the bottom of the screen as in Figure 5-1
- The perf_stats command-line tool in Figure 5-2, which prints core loads to the terminal window.

Each of these have the same default update rate of 2 seconds. The tiperfoverlay plugin adds overhead to DDR and CPU for drawing information to the output frame. Out of box demos use tiperfoverlay by default. Note that the "MPU1_0" metric refers to the average CPU load on the A53 CPU cores.

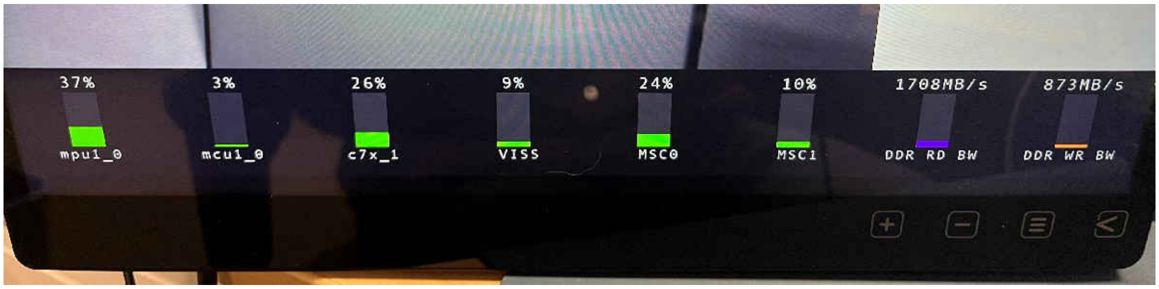
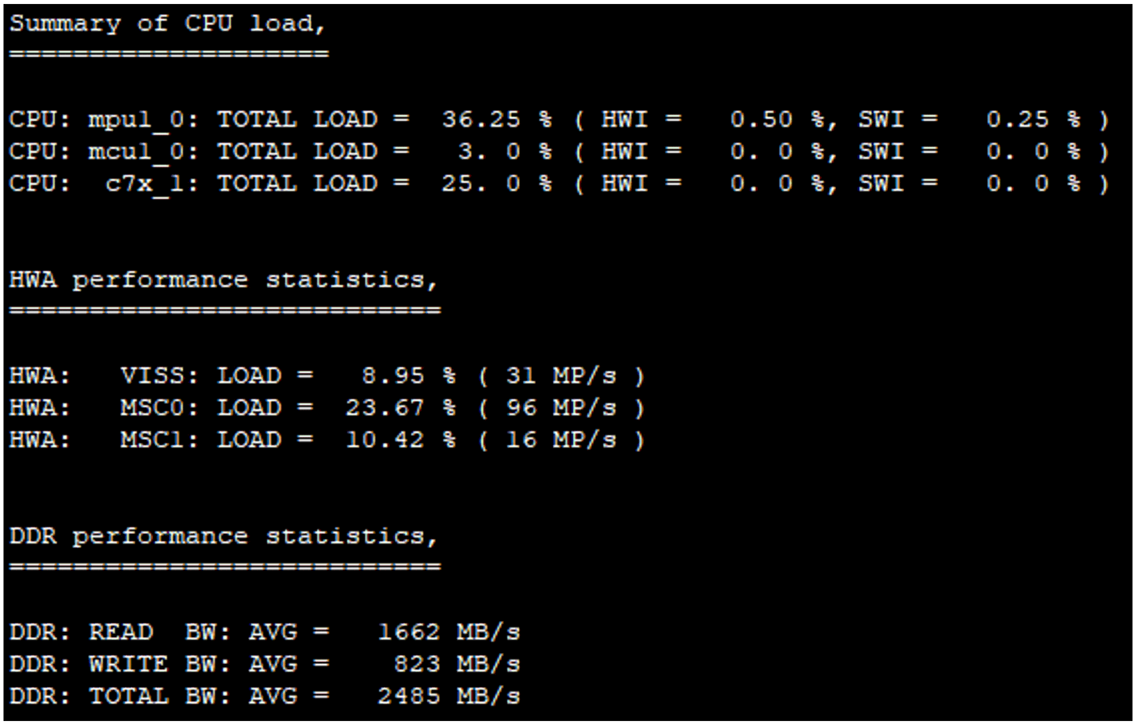


Figure 5-1. tiperfoverlay Gstreamer Plugin When Running the Retail Checkout Demo



```
Summary of CPU load,
====================

CPU: mpu1_0: TOTAL LOAD =  36.25 % ( HWI =   0.50 %, SWI =   0.25 % )
CPU: mcu1_0: TOTAL LOAD =   3. 0 % ( HWI =   0. 0 %, SWI =   0. 0 % )
CPU:  c7x_1: TOTAL LOAD =  25. 0 % ( HWI =   0. 0 %, SWI =   0. 0 % )


HWA performance statistics,
===========================

HWA:   VISS: LOAD =   8.95 % ( 31 MP/s )
HWA:   MSC0: LOAD =  23.67 % ( 96 MP/s )
HWA:   MSC1: LOAD =  10.42 % ( 16 MP/s )


DDR performance statistics,
===========================

DDR: READ  BW: AVG =   1662 MB/s
DDR: WRITE BW: AVG =    823 MB/s
DDR: TOTAL BW: AVG =   2485 MB/s
```

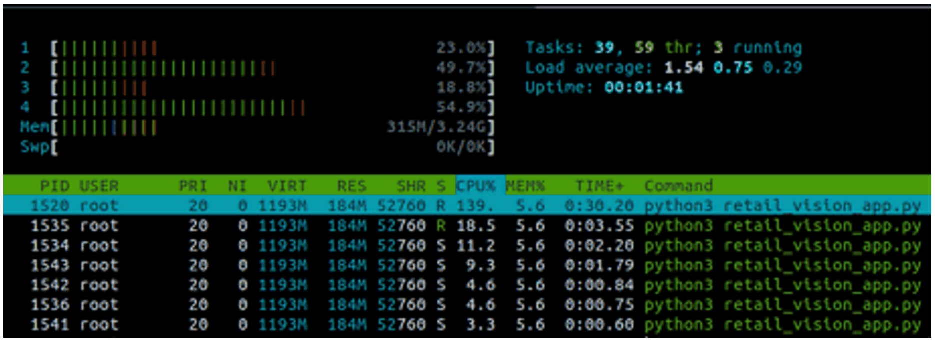
Figure 5-2. perf_stats Command Line Core Load When Running the Retail Checkout Demo

Copyright © 2023 Texas Instruments Incorporated

There is slight variation between these two plots. For consistency, consider the A53 average load to be 35%, the C7x core load as 25% (running at max 1.7 TOPS for E2 EVM), ISP (VISS) is 10% and multiscaler engine (MSC) is average 17%. Note that in Figure 5-1, tiperfoverlay adds overhead for drawing performance information to the screen, which has a small impact on DDR usage.

For the MSC, the overall usage exceeds VISS because the vision pipeline has a split and merge; the performance impact is the summation of the inputs and outputs, which is within a few percent of the 115 MP/s expected for this gstreamer pipeline running at approx. 15 fps. The VPAC accelerator internally contains both MSC and the ISP (VISS); for this analysis consider overall VPAC usage to be 10%.

The DDR is 3200 MT/s and 32-bit, so total 2470 MB/s equates to roughly 20% usage, of which 30% are write transfers. To see how much memory is currently used within userspace, the 'htop' linux utility (see Figure 5-3) is helpful. From this, only around 315 MB is used within the OS. Note the 3.24 GB maximum – this is what remains of the 4 GB LPDDR module on the EVM after carveouts for the Linux kernel and HW accelerators. For example, the C7xMMA is allocated 256 MB by default in the 8.6 Processor SDK Linux. Given this memory usage, a cost-optimized system can use 1.5 GB of DDR, perhaps as low as 1 GB with more optimizations to the Linux image to remove unused services and packages.



```
  1  [||||||||||                          23.0%]   Tasks: 39, 59 thr; 3 running
  2  [|||||||||||||||||||||||             49.7%]   Load average: 1.54 0.75 0.29
  3  [|||||||||                           18.8%]   Uptime: 00:01:41
  4  [||||||||||||||||||||||||||          54.9%]
  Mem[||||||||||                      315M/3.24G]
  Swp[                                     0K/0K]

   PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  1520 root       20   0 1193M  184M 52760 R 139.  5.6  0:30.20 python3 retail_vision_app.py
  1535 root       20   0 1193M  184M 52760 R 18.5  5.6  0:03.55 python3 retail_vision_app.py
  1534 root       20   0 1193M  184M 52760 S 11.2  5.6  0:02.20 python3 retail_vision_app.py
  1543 root       20   0 1193M  184M 52760 S  9.3  5.6  0:01.79 python3 retail_vision_app.py
  1542 root       20   0 1193M  184M 52760 S  4.6  5.6  0:00.84 python3 retail_vision_app.py
  1536 root       20   0 1193M  184M 52760 S  4.6  5.6  0:00.75 python3 retail_vision_app.py
  1541 root       20   0 1193M  184M 52760 S  3.3  5.6  0:00.60 python3 retail_vision_app.py
```

Figure 5-3. Retail Checkout Demo Load Using 'htop' Linux Utility

6 Part Selection

AM62A SoC includes several variants for cost-optimization. The two primary selections to make are in regard to deep learning and CPU performance in Table 6-1.

Table 6-1. AM62A SoC Variants Based on Deep Learning Performance and CPU Core Count

Part	C7xMMA (TOPS)	A53 (#)
AM62A74	2	4
AM62A72	2	2
AM62A71	2	1
AM62A34	1	4
AM62A32	**1**	**2**
AM62A31	1	1

Considering table and the loading information, the AM62A**32** is sufficient for this retail checkout application. This SoC variant has 1 TOPS of performance by down-clocking the accelerator from 1 GHz to 500 MHz, of which only 50% is needed for this application running at 15 fps. The dual core variant is also acceptable given the average load is 35% across four CPU cores.

7 Power Usage

The AM62A is built with low-power applications in mind. There is a *AM62Ax Power-Estimation Tool (PET)* [4] available on the AM62A product page. Core loading and clock frequencies are used as parameters to estimate power based on benchtop measurements collected from bare-metal tests, which implies there is no operating system like Linux. To emulate an AM62A32 in this power estimation tool, the C7x frequency should be 500 MHz or less, and only two of the A53 cores should be above 0% utilization. To obtain the most accurate power estimations, see the latest version of the PET.

Copyright © 2023 Texas Instruments Incorporated

Benchtop measurements collected for the retail-scanner application power usage show average power consumption of 1810 mW at a junction temperature of 42°C. The processor's configuration used 500 MHz for the C7xMMA frequency and 1250 MHz for the CPU frequency.

This power measurement is at least 300 mW higher than an optimized processor configuration. The 8.6 Linux SDK enables and clocks all components across the processor, whether they are being utilized or not. Therefore, power consumption on the Linux SDK at product launch is higher than a device configuration that is optimized for low-power.

8 Summary

The AM62A is well-suited for vision-based retail systems requiring advanced image processing at the edge. In this document, a demo application for an automated retail checkout system is analyzed to determine the core load across the heterogenous AM62A processor architecture. This information is then used to select a suitable cost-down variant of the processor and an estimate of the power consumption.

9 References

1. "Model Analyzer," [Online]. Available: https://dev.ti.com/edgeaisession/. [Accessed 26 April 2023].
2. "Edge AI Retail Checkout Demo," 15 April 2023. [Online]. Available: https://github.com/TexasInstruments/edgeai-gst-apps-retail-checkout/blob/main/retail-shopping. [Accessed 26 April 2023].
3. Texas Instruments: *Building an Edge AI Application for Automated Retail Scanner on AM6xA MPUs*
4. Texas Instruments: *AM62Ax Power-Estimation Tool (PET)*

Copyright © 2023 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023, Texas Instruments Incorporated