Final Project Proposal
HDR Image Signal Processor (ISP)

NTHU/Team 7

Leader:張傑閔

Member:蔡宗頴

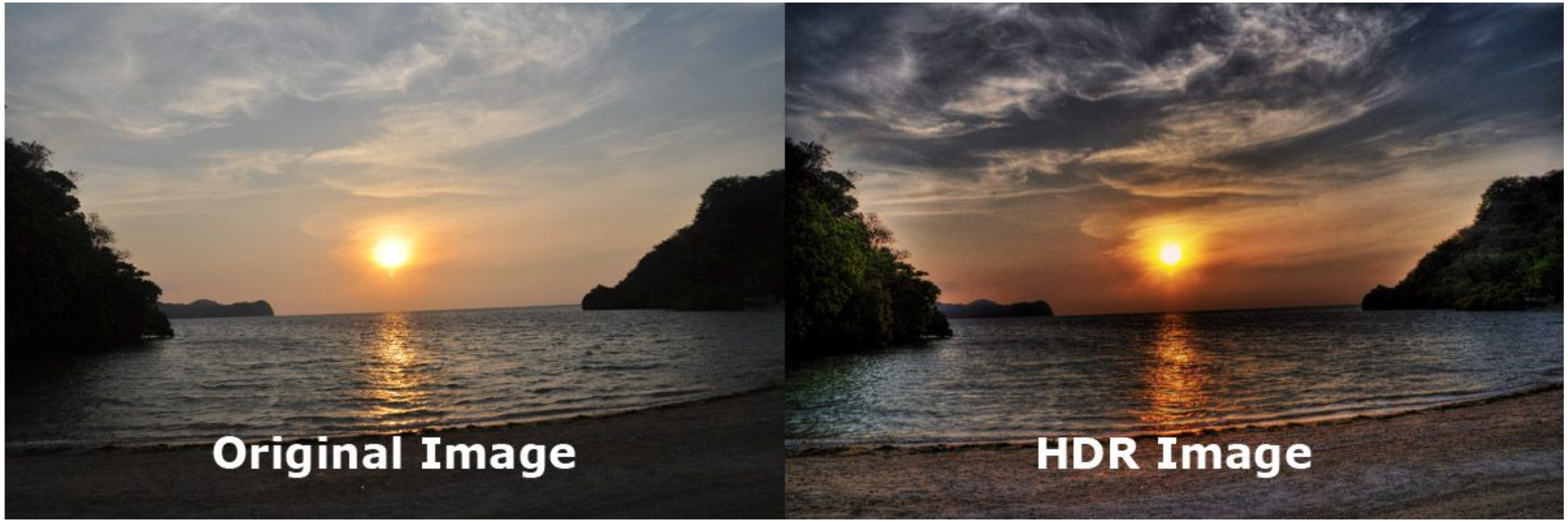
Outline

- Problem Statement
- Problem Scope
- Project Plan
- Reference

Problem Statement

Application

- Application :
  - Preserve the true essence of your surroundings with HDR imaging techniques.



This image is copied from: https://vmi.tv/blog/learn-help/hdr_reality_and_monitoring-a_dops_perspective/

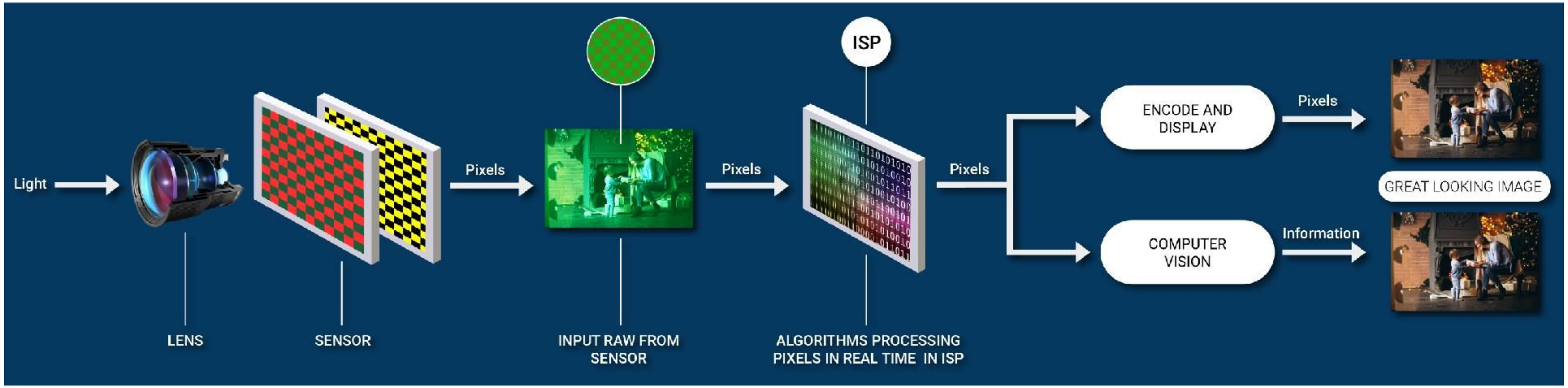
Issue and Objective

- Issue:
    - Limited sensor dynamic range sacrifices image detail in either bright or dark areas, resulting in an incomplete representation of the scene.
- Objective:
    - The target to achieve is the function of real-time HDR ISP.

Problem Scope

Background Introduction

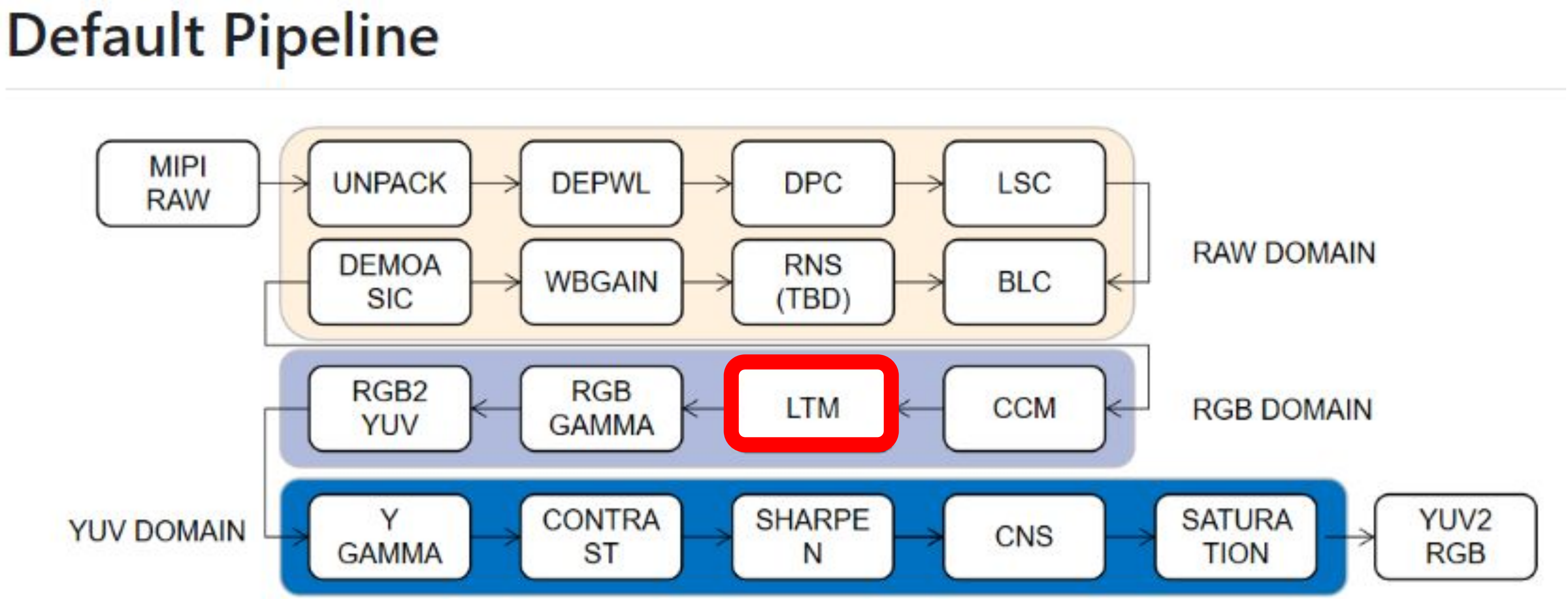
- Where is ISP used?
  - ISP: Image Signal Processor.
  - Image Signal Processing is a method to convert an image into digital form by performing operations for digital processing and image quality enhancement, and dedicated hardware called an ISP implements these functions.



This image is copied from: https://www.faststreamtech.com/blog/image-signal-processor/

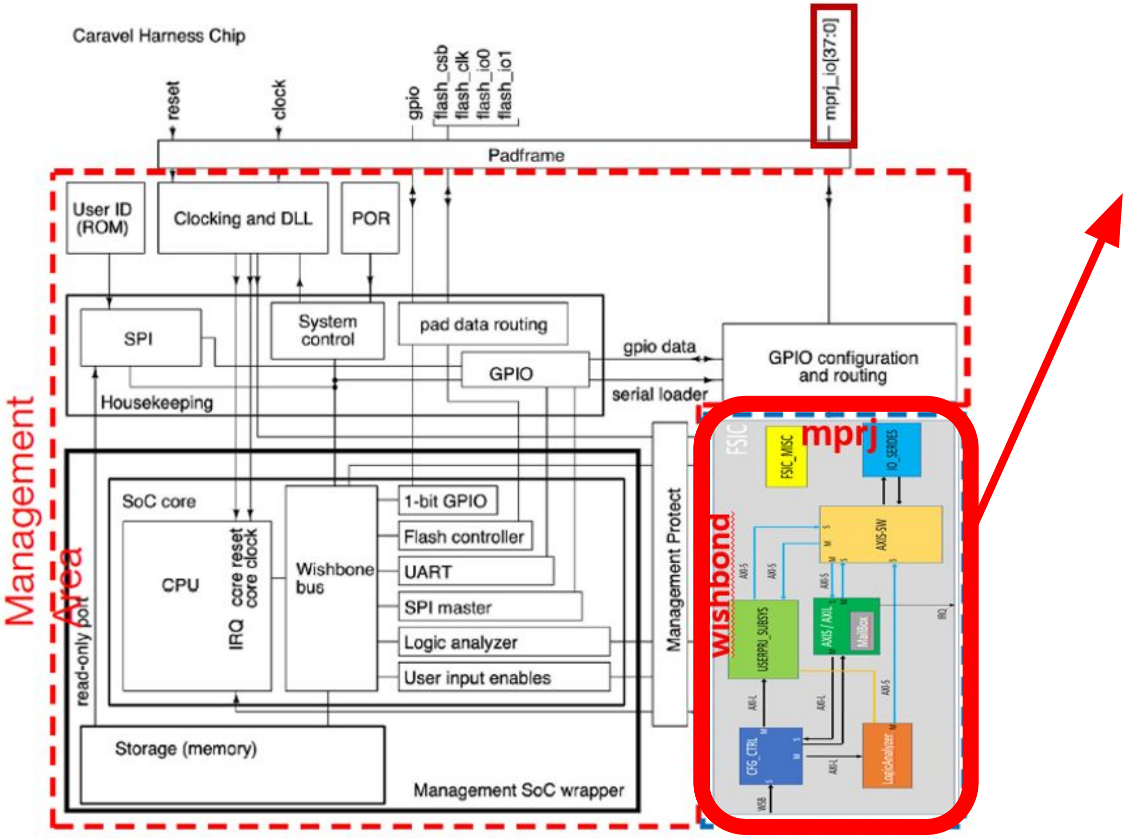
Background Introduction

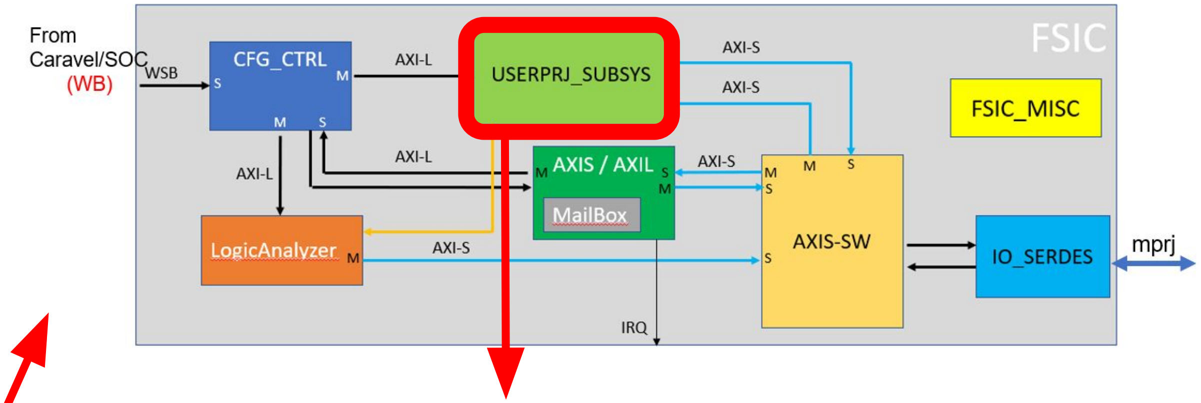
- When is HDR introduced?
  - LTM: Local Tone Mapping
  - Purpose: Local contrast enhancement based on brightness domain.



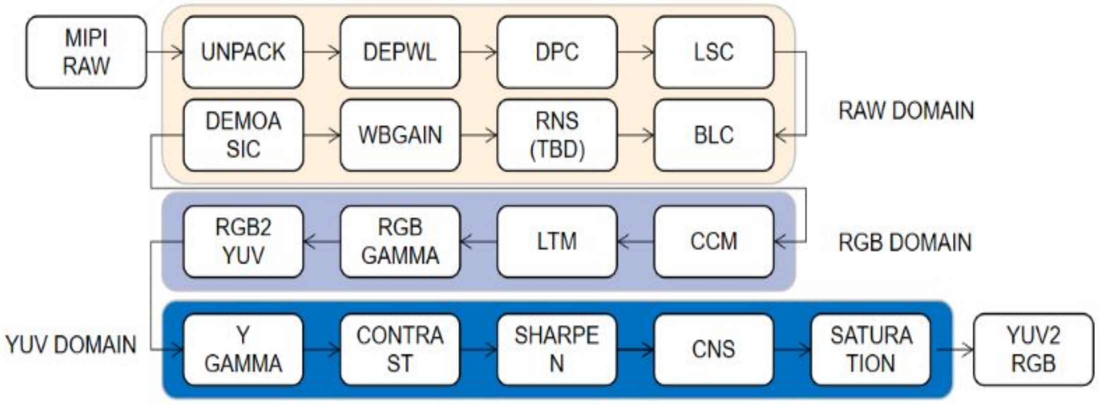
System Block Diagram and Its Operation Flow

- Block diagram





HDR ISP (Operation Flow)



©BOLEDU

Identify the area of work

- Identify algorithm C-source code and run Catapult C-sim
  - Self-contained, no library function call
  - Identify test dataset
    - Generate raw images or copy some raw images from the internet.
    - The number of test pictures needed are generated according to the specification.
  - Partition host + kernel
    - Host: Send configurations
    - Kernel: HDR ISP pipeline
  - Run Catapult C-sim (pre-HLS verification)

Identify the area of work

- Kernel HLS implementation, Host implementation
  - Kernel HLS implementation for 17 modules in pipeline.
  - Unit test for every module in Catapult (RTL verification).
  - Host implementation for sending configurations.
- Individual Kernel FPGA validation/integration test
  - FSIC simulation and validation by testbench.
- Kernel and Host Optimization
  - If time permits, optimize at least one module to enhance performance or achieve better quality of results.

FPGA platform

- KV260

Target Specification

- Primary camera input format: MIPI RAW10/RAW12/RAW16
- Max. resolution: (MIPI RAW) 1280 x 960 (1.2 Mp)
- Max. throughput: RGB pixels per cycle: 1 RGB pixel (24-bit) per cycle
  - 45 fps @ 1.2 Mp, 60 fps @ 720p
- Max. frequency: 200 MHz

Project Plan

Workflow

- Identify algorithm C-source code and run Catapult C-sim - 1w
  - Self-contained, no library function call (Done, tested with Catapult Makefile)
  - Identify test dataset - 3 days
    - Generate raw images or copy some raw images from the internet.
    - The number of test pictures needed are generated according to the specification.
  - Partition host + kernel (Done)
    - Host: Send configurations
    - Kernel: HDR ISP pipeline
  - Run Catapult C-sim (pre-HLS verification) - 3 days

Workflow

- Kernel HLS implementation, Host implementation - 3w
  - Kernel HLS implementation for 17 modules in pipeline.
  - Unit test for every module in Catapult (RTL verification).
  - Host implementation for sending configurations.
- Individual Kernel FPGA validation/integration test - 2w
  - FSIC simulation and validation by testbench.
- Kernel and Host Optimization - Max. 1w
  - If time permits, optimize at least one module to enhance performance or achieve better quality of results.

Roles & Task assignment

- 張傑閔:
- Identify algorithm C-source code and run Catapult C-sim
    - Run Catapult C-sim
- Kernel HLS implementation, Host implementation
    - Modules: LTM to YUV2RGB
- Individual Kernel FPGA validation/integration test
    - FSIC simulation

- 蔡宗頴:
- Identify algorithm C-source code and run Catapult C-sim
    - Identify test dataset
- Kernel HLS implementation, Host implementation
    - Modules: UNPACK to CCM
- Individual Kernel FPGA validation/integration test
    - FSIC validation

Check-point, Time Duration, and Deliverables

Date	Check-point	Time duration	Deliverables
5/5	Identify algorithm C-source code and run Catapult C-sim	1w	1. Test dataset 2. C-sim result
5/12	Kernel HLS implementation, Host implementation (I)	1w	1. # of modules implementation & unit test done
5/26	Kernel HLS implementation, Host implementation (II)	2w	1. # of modules implementation & unit test done
6/9	Individual Kernel FPGA validation/integration test	2w	1. FSIC simulation pass or not 2. FSIC validation pass or not
6/16	Kernel and Host Optimization	1w	1. Performance speedup or QoR

Reference

List of Papers for Reference

[1]B. C. Huang, C. S. Fuh, “Image pipeline algorithms for standard mobile imaging architecture sensors”, 2005 18th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP), pp. 1118-1125, 2005.
[2]S. H. Choi, et al. “A parallel camera image signal processor for SIMD architecture”, EURASIP Journal on Image and Video Processing, vol 2016, pp.1-14, 2016.
[3]Park H.S. (2016) Architectural Analysis of a Baseline ISP Pipeline. In: Kyung CM. (eds) Theory and Applications of Smart Cameras. KAIST Research Series. Springer, Dordrecht.

Open-source to use

- We use the following open-source projects:
  - Baseline: Image Signal Process For HDR CMOS Image Sensor
    - We follow the HDR ISP pipeline in this repository.
  - Catapult HLS Implementation reference: xkISP
    - Both the Vitis and Catapult HLS C modules are provided for reference, but the algorithm C codes are encrypted and provided in the form of executable binary file.

Thank you!

Final Project Proposal Guidelines

Final Project (Team)

- Refer to “Referenced Final Projects” for suggested projects
  - https://docs.google.com/spreadsheets/d/1FqAnTJP_vOm9G4UFEAlNB9KhNTy6tggH/edit?usp=drive_link&ouid=106716318998274820333&rtpof=true&sd=true
- Your research topics
- Requirement: End-to-end application acceleration
  - Scope at application level
  - Profiling to identify function to accelerate
  - SW application parallelization
  - Kernel optimization
  - Run on FPGA

Start thinking the Final Project Now

Purpose

The proposal report serves

1. Provide a framework to construct a project, from idea to execution
2. Define the project scope and evaluate its complexity
3. Basis for one-on-one discussion

Content of Final Project Proposal

- Project Title
- Team: Leader + Members
- Problem statement
- Project scope
- Project plan
- Reference

Problem Statement

- Context: what is the application the project is applied to?
- Issue: what is the problem to solve?
- Objective: what is the target to achieve? e.g. performance/area improvement, adding new functions ...

Project Scope

- Background Introduction
- System block diagram, and its operation flow
- Identify the area of work
- What FPGA platform to implement on, e.g. U50, PYNQ-Z2, KV260
- Target Specification – throughput, accuracy, speedup, area, or a design methodology proposed

Project Plan

- Workflow definition: Breakdown the project into a set of tasks, and describe the dependency among the tasks
- Assign the task to members / define the role of each member
- Identify the check-point, estimate the time duration, deliverables, and quality requirement

Workflow

- Identify algorithm C-source code - 1w
    - self-contained, no library function call
    - Identify test dataset
    - Partition host + kernel
- Run C-sim in Vitis environment Partition - 2w
    - run through dataset -> check correctness
- Kernel HLS implementation, Host implementation - 2w
    - define host/kernel communication, including debugging
    - If multiple kernels, allow validate separately.
    - Host program implement two modes for each kernels (C-code, or FPGA kernel)
- Individual Kernel FPGA validation/integration test - 1w
- Kernel and Host Optimization