QR Code Scanner

6.2050 Final Project, Fall 2023

M.Subhi Abo Rdan, Ayana Alemayehu

What is a QR Code?

- QR Codes stand for Quick Response Codes, used to visually represent data like text, images, pure bytes, etc
- Black is 1, White is 0, each little square called a module
- Version number dictates QR Code size (Version 1 is 21x21)
- Split into Model 1 (max 73 x 73) and improved Model 2 (177 x 177)
- Contains alignment patterns and error correction capabilities for software to robustly decode information despite unfavorable conditions (shadows, scratches, marks, etc)

System Overview

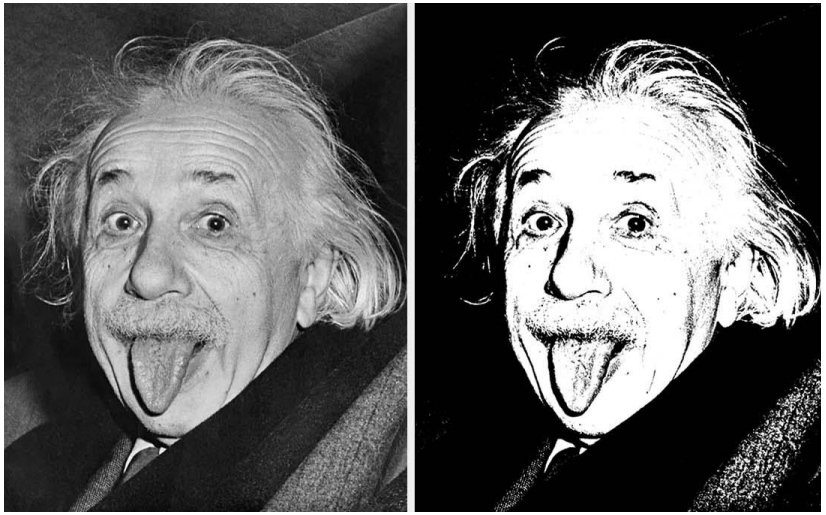
- Build a QR code reader that is relatively robust to translations and rotations of the original QR code.
- Fixed camera setup and well lit
- Output will be dumped to registers that are read through Manta

changes since block diagram highlighted

QR Code Reader: Approach

Image Grabbing

- 480 x 480 photo taken by the camera module
  - Resolution is maximum square output of the camera
- Image binarized upon arrival to BRAM determined by threshold



Block Diagram, Stage 1

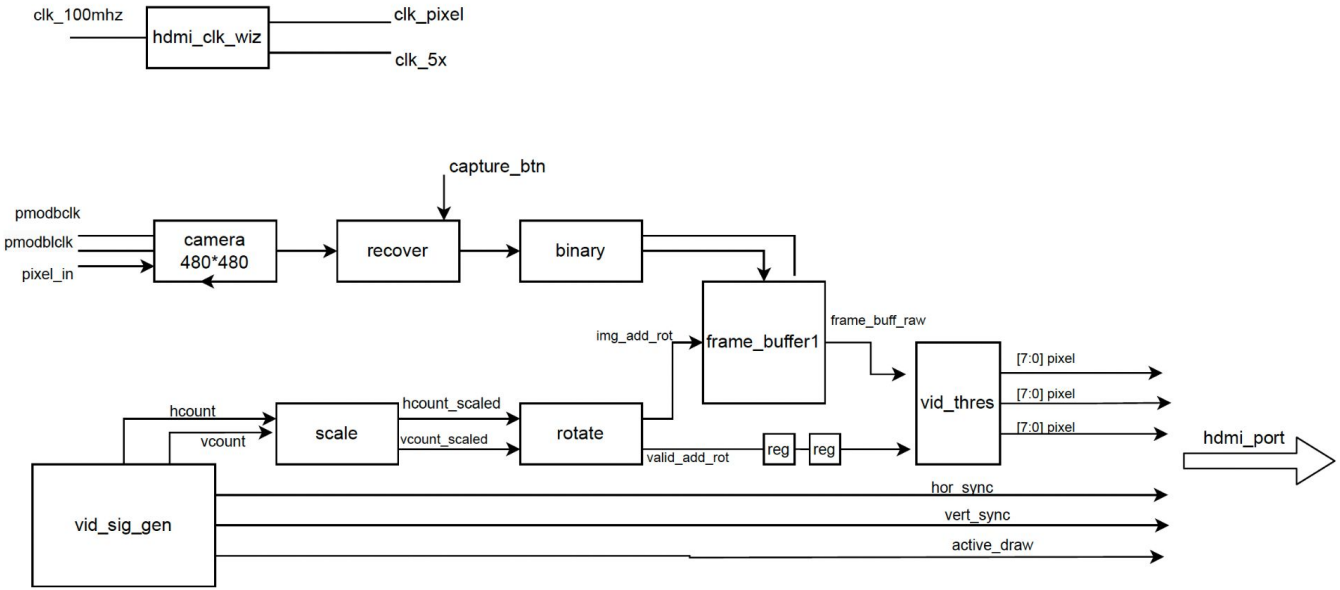
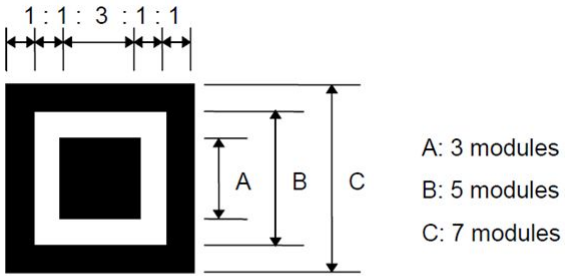


Image Manipulation & QR Code Detection

- Convolution(s) applied to remove unwanted features to image
  - Sharpen kernel
  - “Mode” kernel to average pixel values depending on neighbor
- Finder Features detected through 1:1:3:1:1 pixel ratio

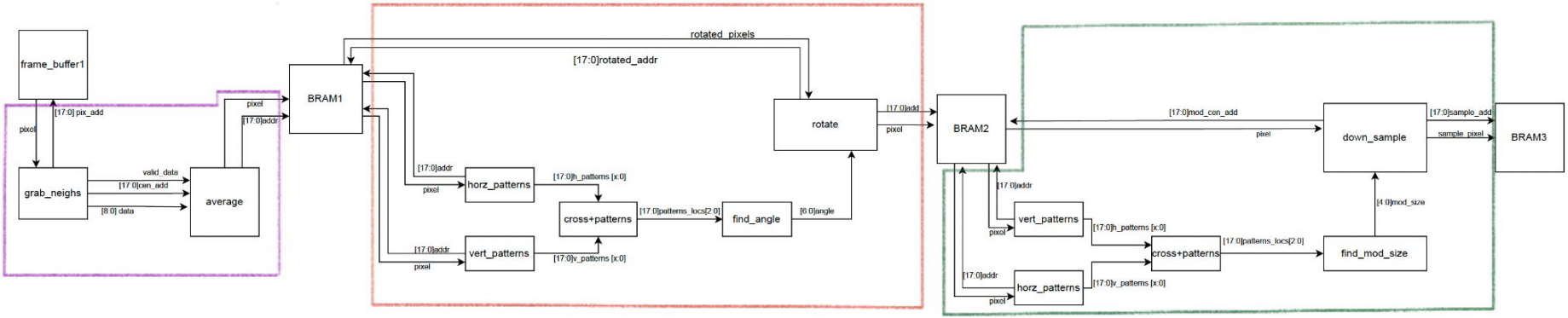


Structure of a finder pattern

- Rotation of QR code detected through lines going through finder features
- Un-done via rotation matrix

Reference number ISO/IEC 18004:2000(E)

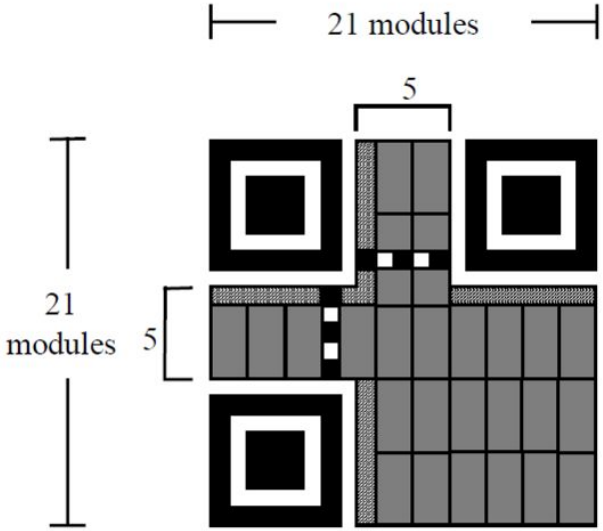
Block Diagram, Stage 2



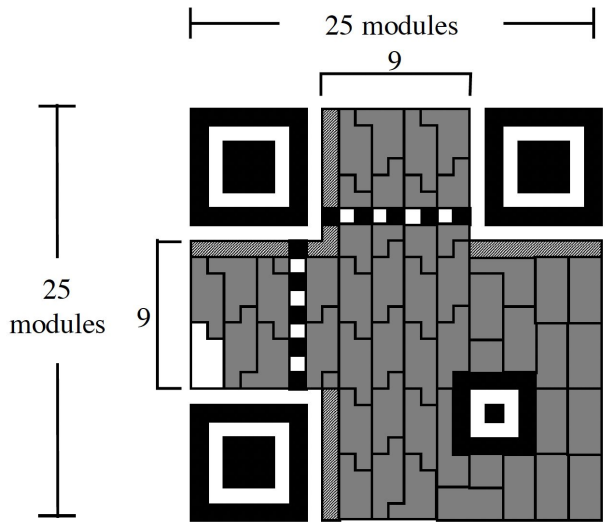
Decoding

QR Code Versions

Will start with version 1



Version 1

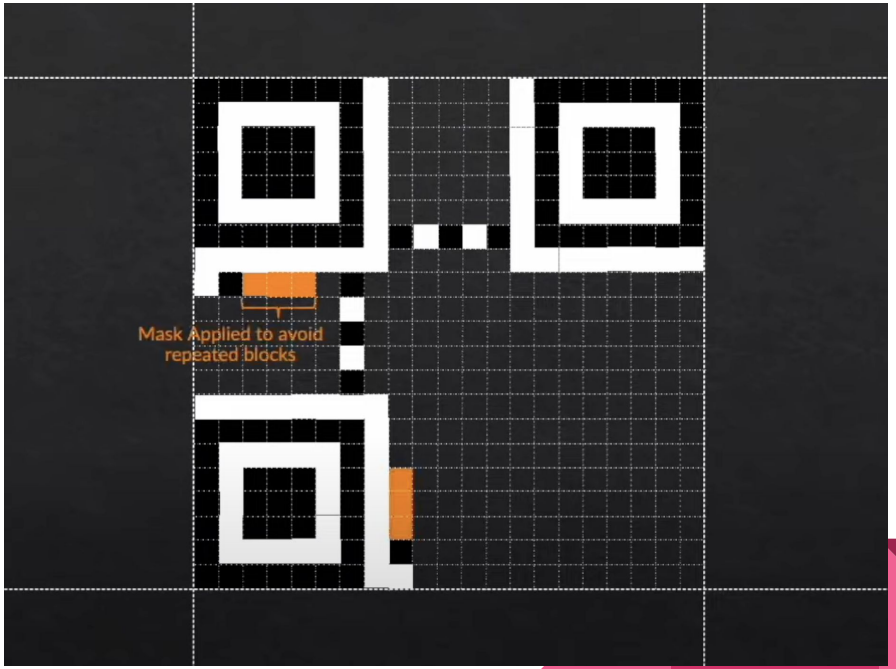


Version 2

Reference number ISO/IEC 18004:2000(E)

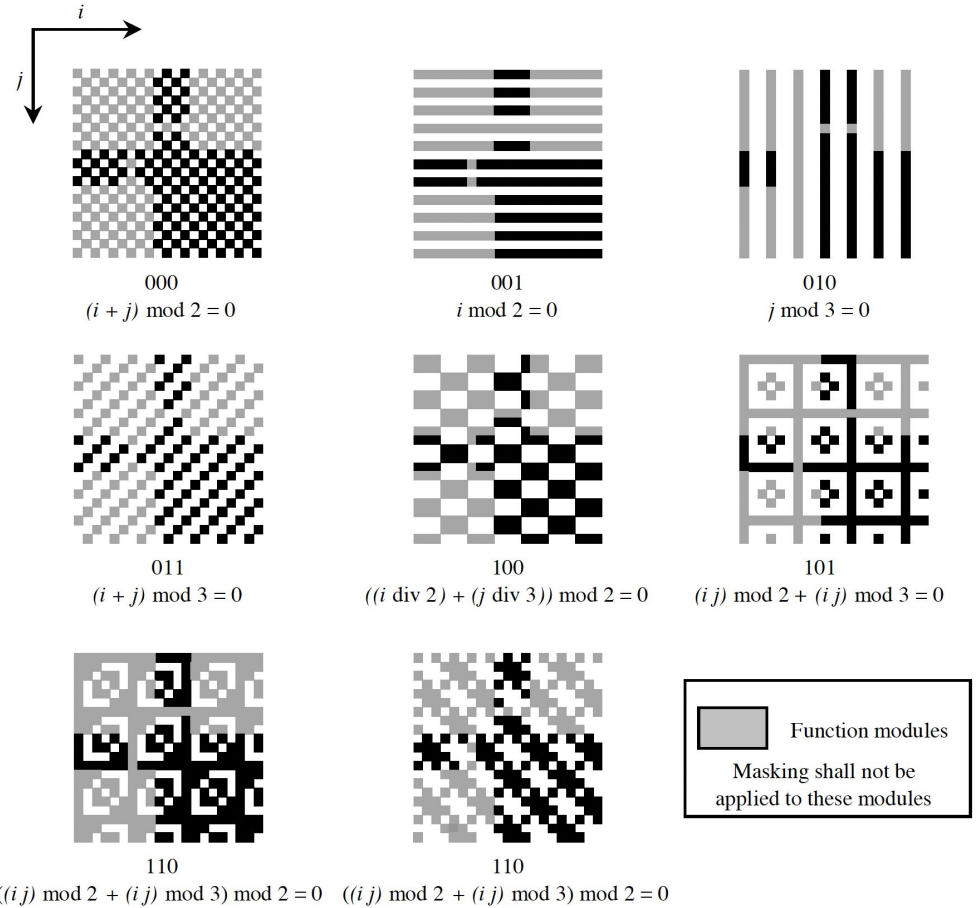
Unmasking the data

- Mask type decoded in these three modules
- 8 kinds of masks, used to prevent malformed qr codes
- Masks are XOR’d with the data, and only applied to data regions



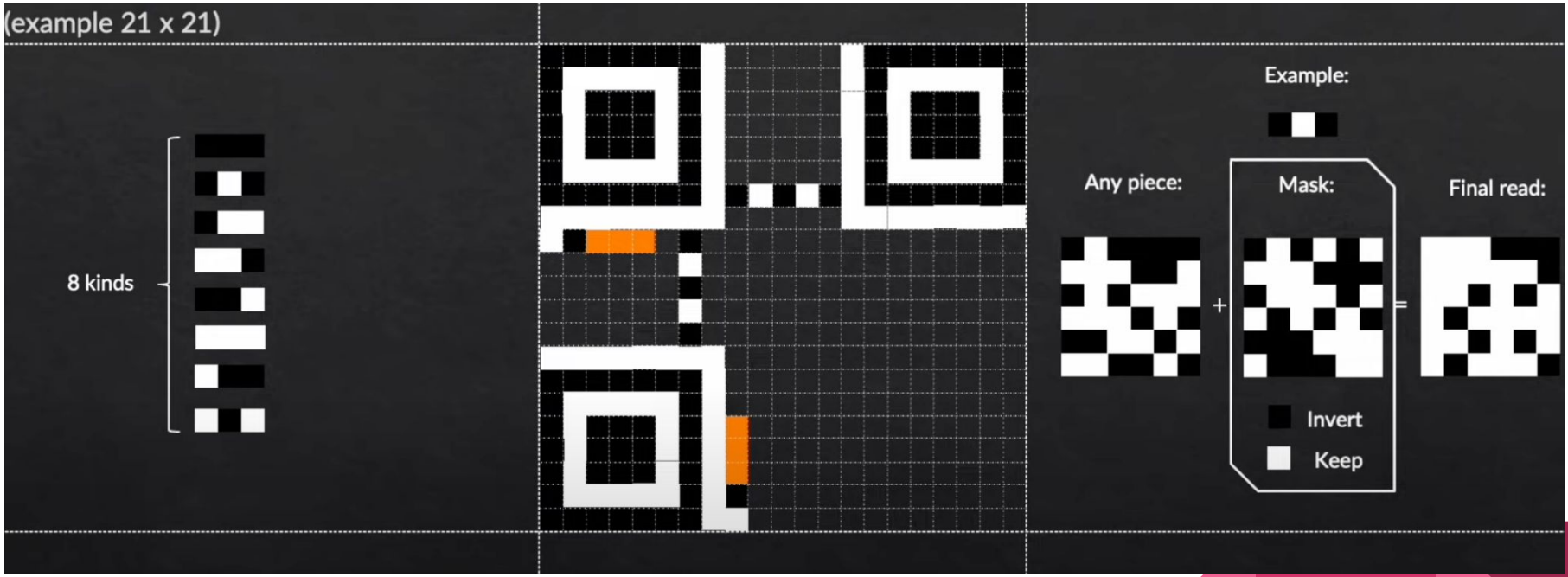
https://www.youtube.com/watch?v=Rc3ul6RRANU

Masks



Reference number ISO/IEC 18004:2000(E)

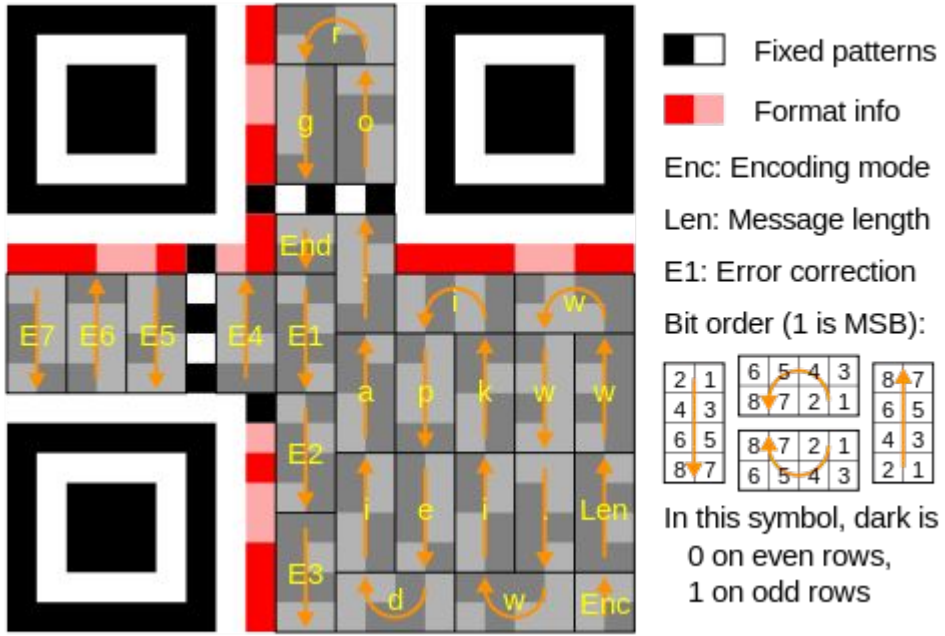
Unmasking the data



https://www.youtube.com/watch?v=Rc3ul6RRANU

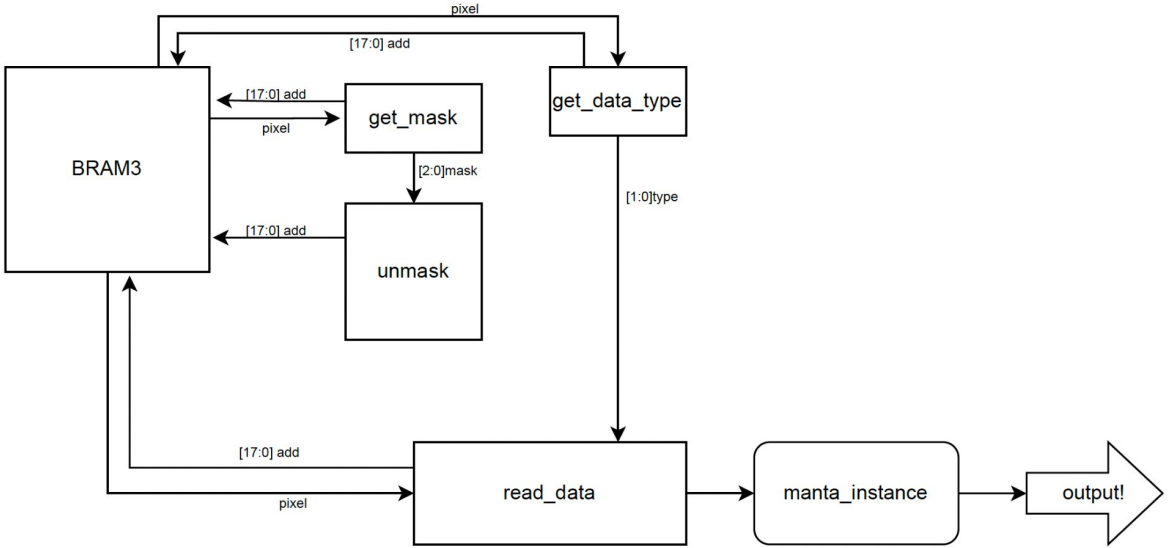
Data Reading

- First byte after encoding bits has the length of the data
- Bit order is different depending on the reading direction
- Final block of data is padded with 0’s if necessary
- Error correction blocks
  - Error correction outside project scope



https://blog.qartis.com/decoding-small-qr-codes-by-hand/

Block Diagram, Stage 3



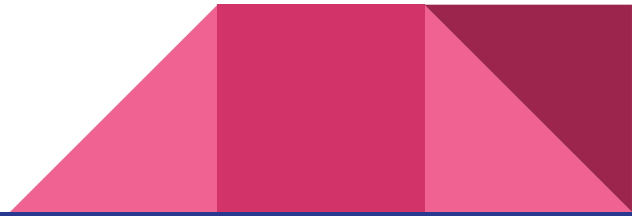
Testing

High-Level Testing

- Testing split by stage
- Stage 1):
  - Live feed of binarization of cameras output through HDMI (720p)
- Stage 2):
  - Multiplex 720p HDMI output between the different BRAMS representing the different stages of the image processing pipeline…
    - BRAM1: Image after average/convolution applied
    - BRAM2: Rotated image
    - BRAM3: Final detected QR code
- Stage 3):
  - Output decoded QR code through manta

Module-Specific Testing

- We will write custom test-benches for each module that reflect real world conditions/inputs, including potential noise and other things where necessary
- For image tests, can write python scripts that convert an image into a testbench



QR Code Scanner Checklist

MINIMUM VIABLE PRODUCT

- QR Code is non-rotated, centered and close to camera when picture is taken
- Picture taken in well lit environment with minimal noise
- QR Code data extraction and decoding accurately converts QR code into its underlying data, reported through manta
- Supports bytes data type

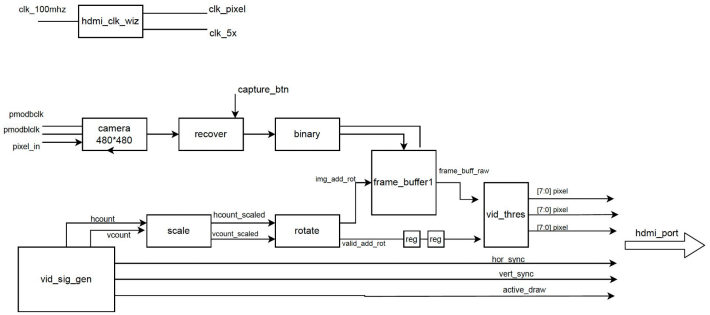
GOAL PRODUCT

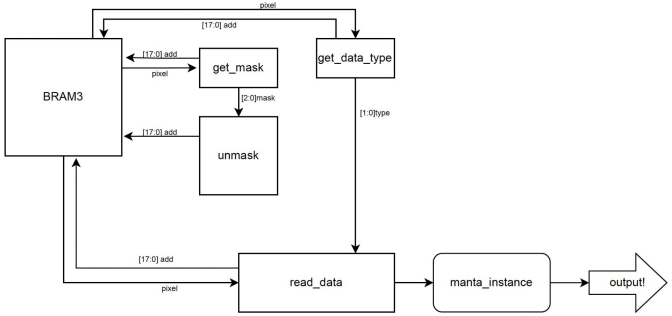
- rotations and translations are supported as well.
- Supports numeric & alphanumeric data types.

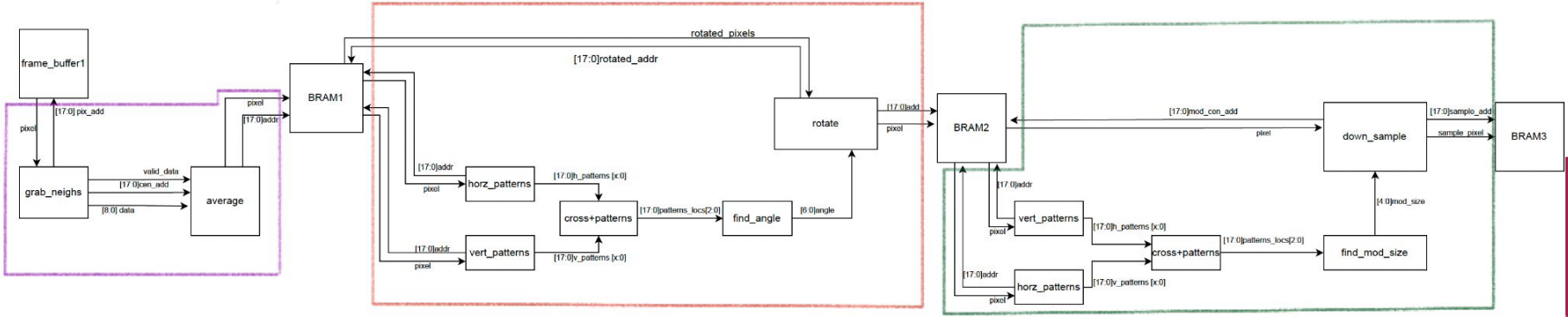
STRETCH GOALS

- Other data types
- Extend to higher QR code versions
- Implement error correction via external IP

Core Modules







Timeline

Nov 7 -> 14	Stage 1 & Testing + Physical Setup + High-Level Module Skeletons + State Machine Skeleton Complete
Nov 14 -> Nov 21	Stage 2 Complete
Nov 21 -> Nov 28	Preliminary Report (on Nov 21-23), Stage 3 + Testing Stage 3
Nov 28 -> Dec 5	Finalize (Any Issues, testing, possibly attempt stretch goals)
Dec 5 -> Dec 13	Final Report + video

Questions?