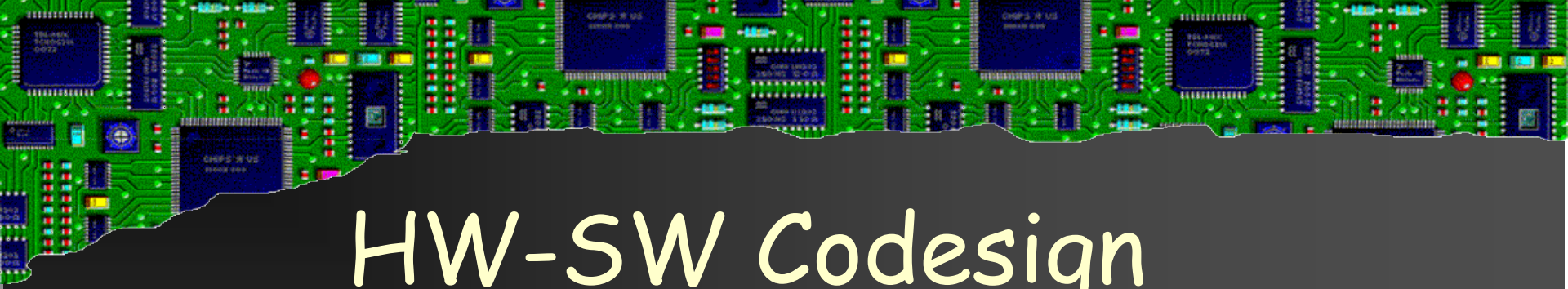


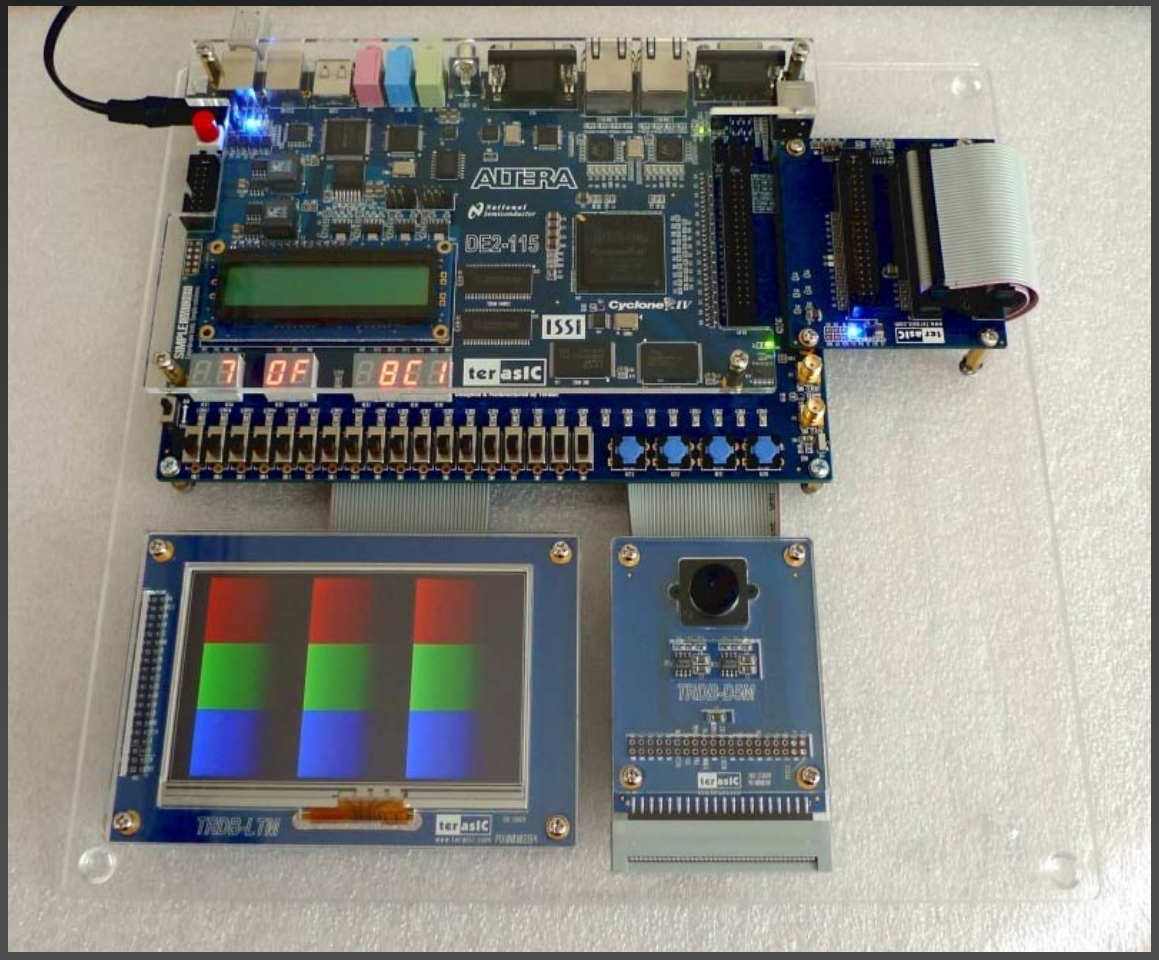
HW-SW Codesign

Lab Course - Excercise

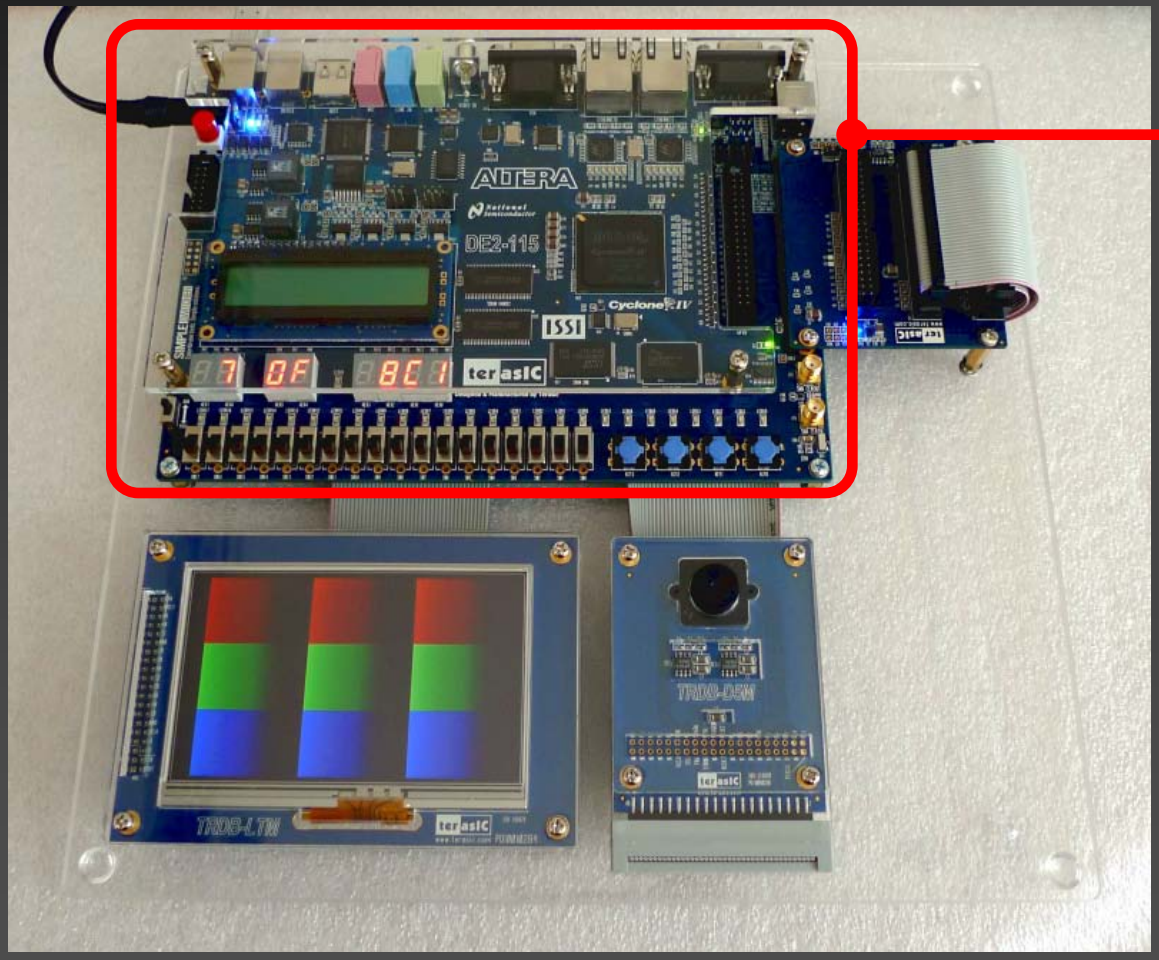
Jakob Lechner,
Thomas Polzer

http://ti.tuwien.ac.at/ecs/teaching/courses/hwswcode_lu

Target Platform



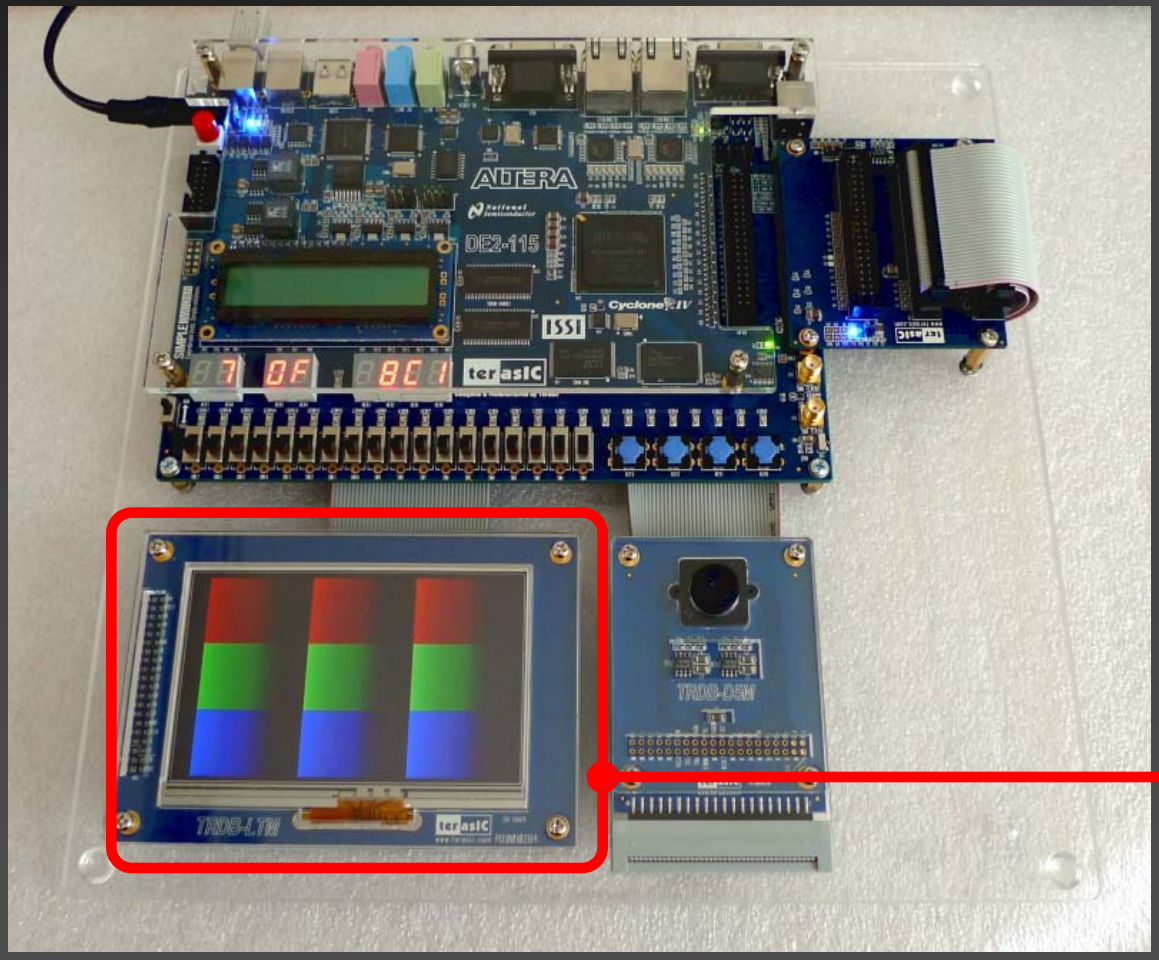
Target Platform



FPGA board

Cyclone IV FPGA
VGA, RS-232, PS/2
2x Ethernet
USB Host/Device
Audio In/Out, Video In
LCD, 7-segments
IR-Receiver
SD-Card Reader
128MB SDRAM
2MB SRAM
8MB FLASH
Switches, Pushbuttons

Target Platform

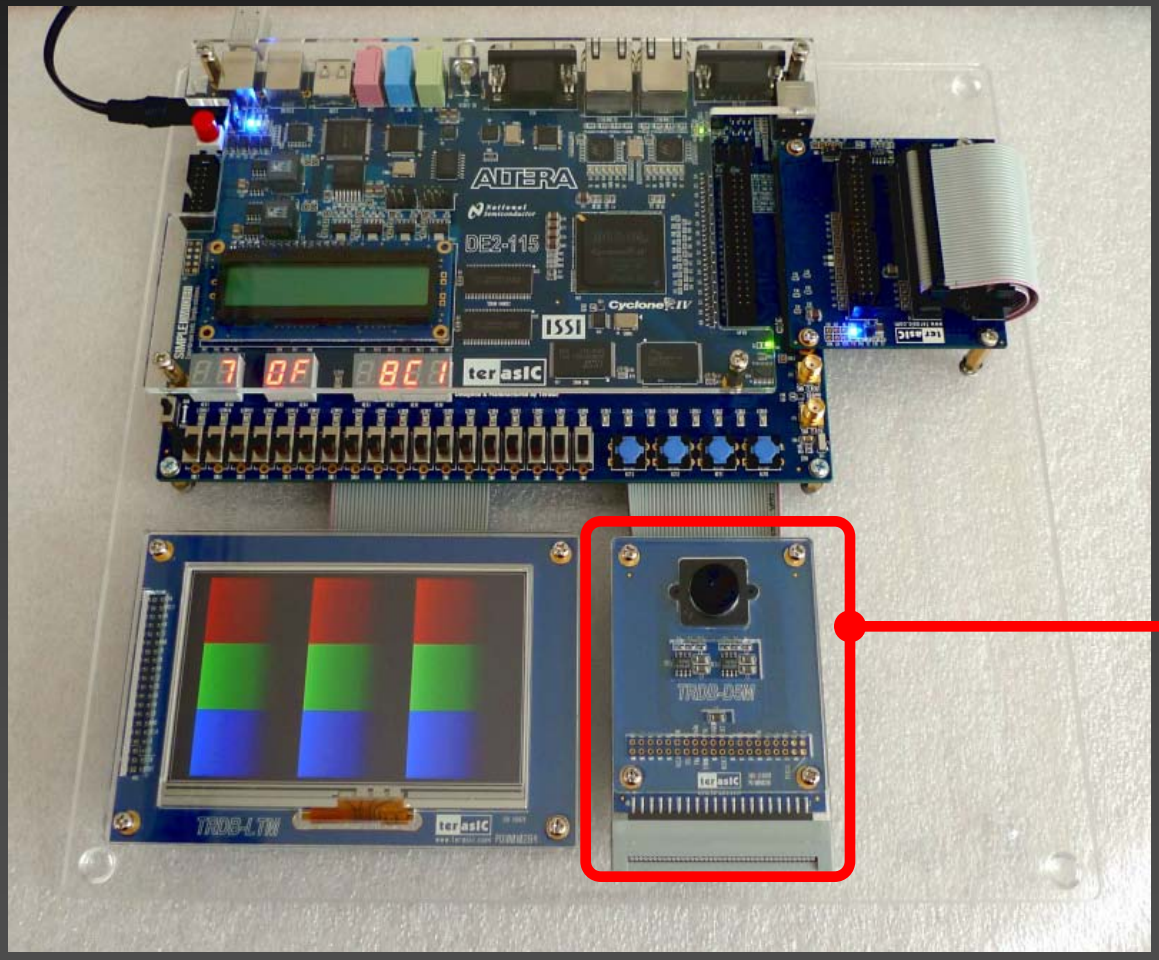


4.3 inch screen
24-bit color depth
max. resolution 800x480

12-bit ADC for
touch coordinates

Touchscreen

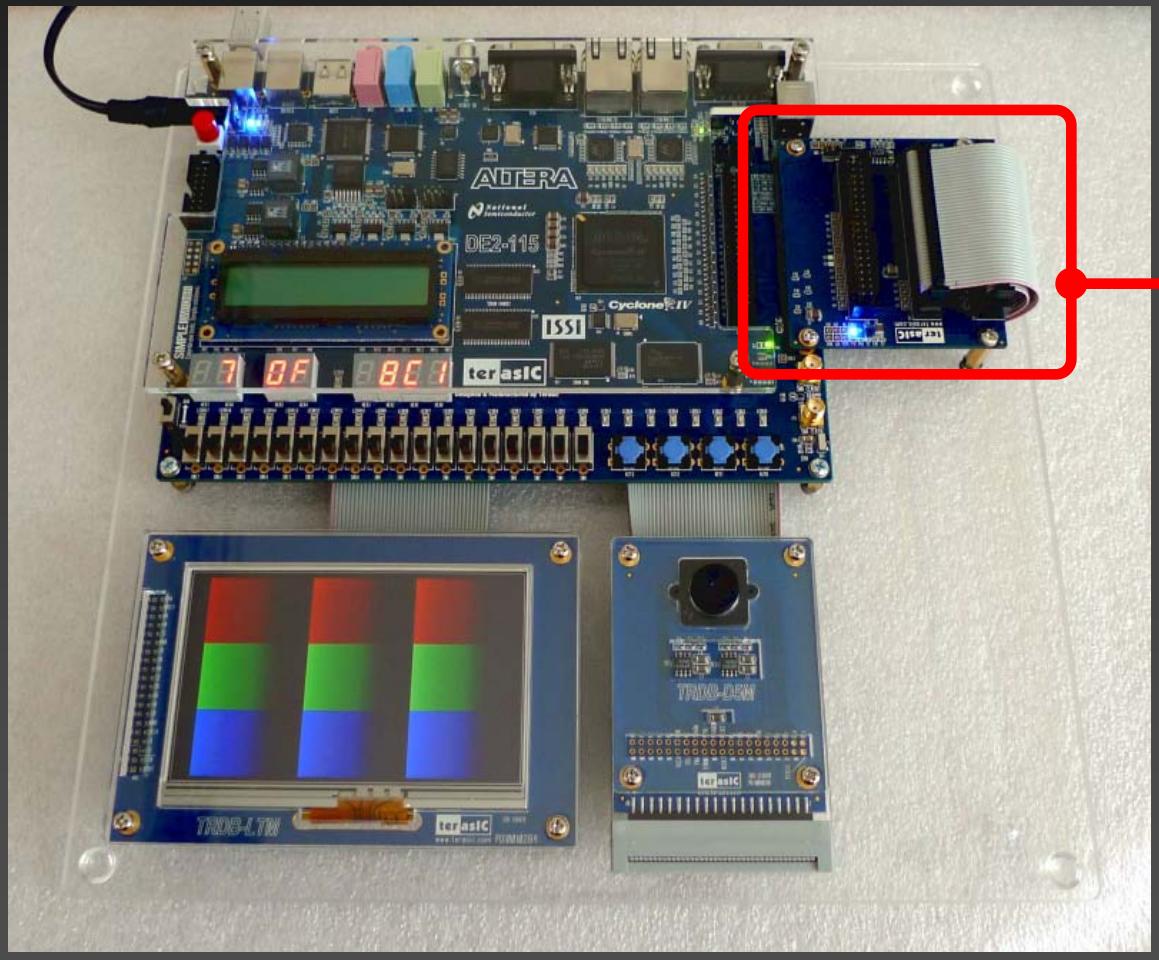
Target Platform



Camera

up to 5 Megapixels
up to 70 fps

Target Platform



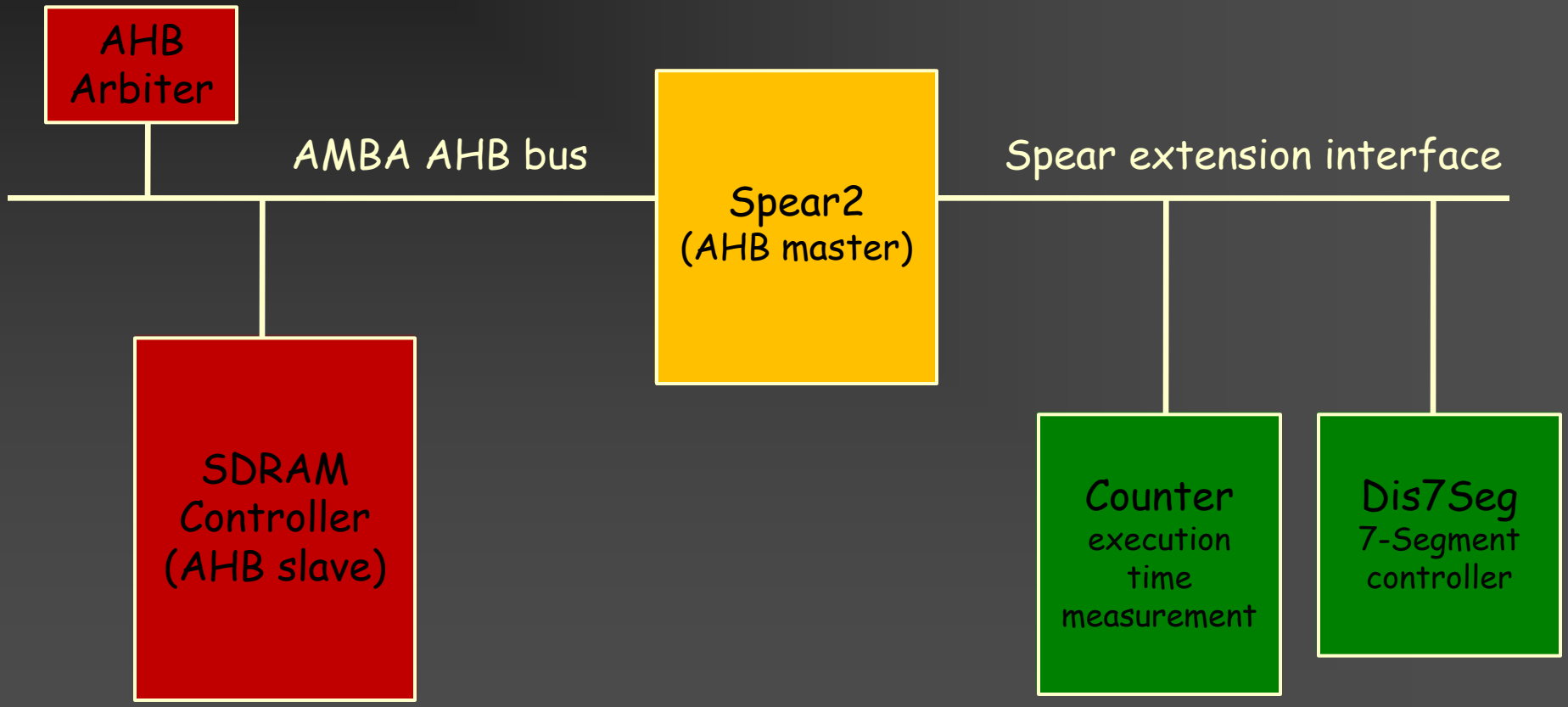
Interface board

3x 40-pin extension connectors

HW/SW-Codesign Setting

- FPGA: Altera Cyclone IV
  - 114480 ALUTs
  - Embedded memory: 3888 Kbits
- Spear2
  - Clock: 50 MHz
  - Instruction Mem: 128K
  - Data Mem: 128K
  - No FPU, no multiplication

HW/SW-Codesign Setting



The Application

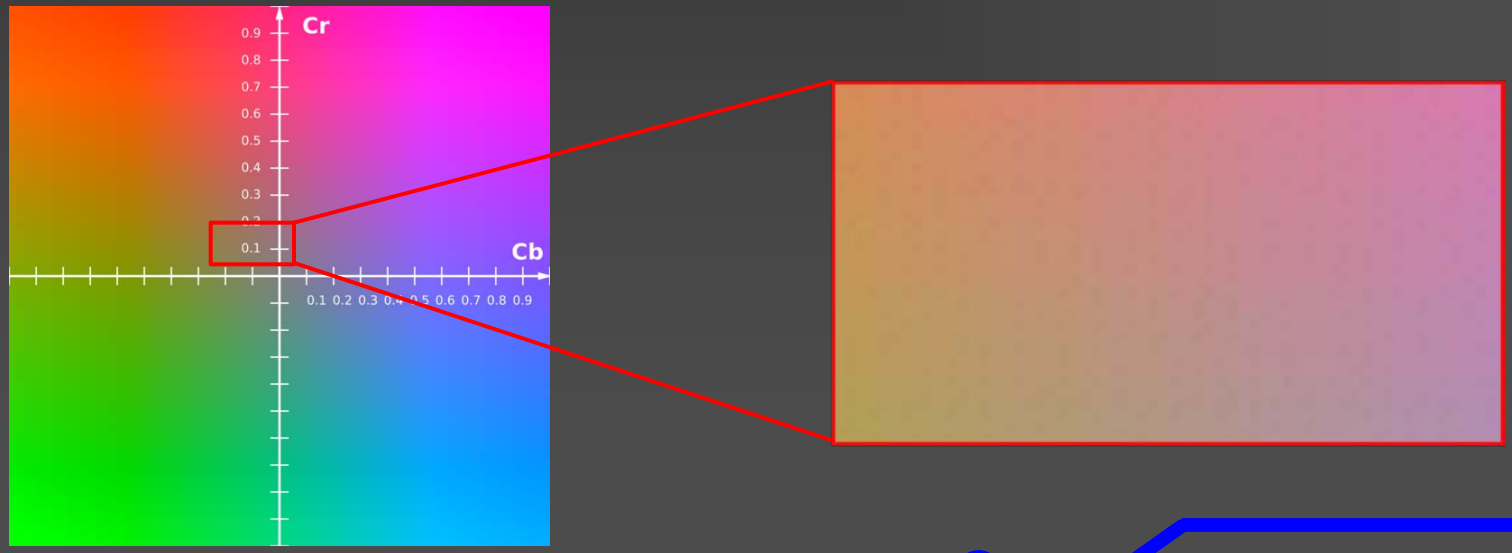
- Fetch video frames from camera
  - Resolution: 640x480 pixel
  - Real-time image processing
    - Frame rate: 20-30 fps
- Perform face detection
  - Face detection = find position of face in an image
  - Draw rectangle around detected face
- Output processed images on touchscreen

Face Detection

- Simple algorithm based on finding skin-colored regions
- Processing steps (input: RGB image):
  1. Conversion to YCbCr color space
  2. Apply skin filter
  3. Erosion filter for reducing noise
  4. Dilation filter
  5. Image segmentation
  6. Compute coordinates

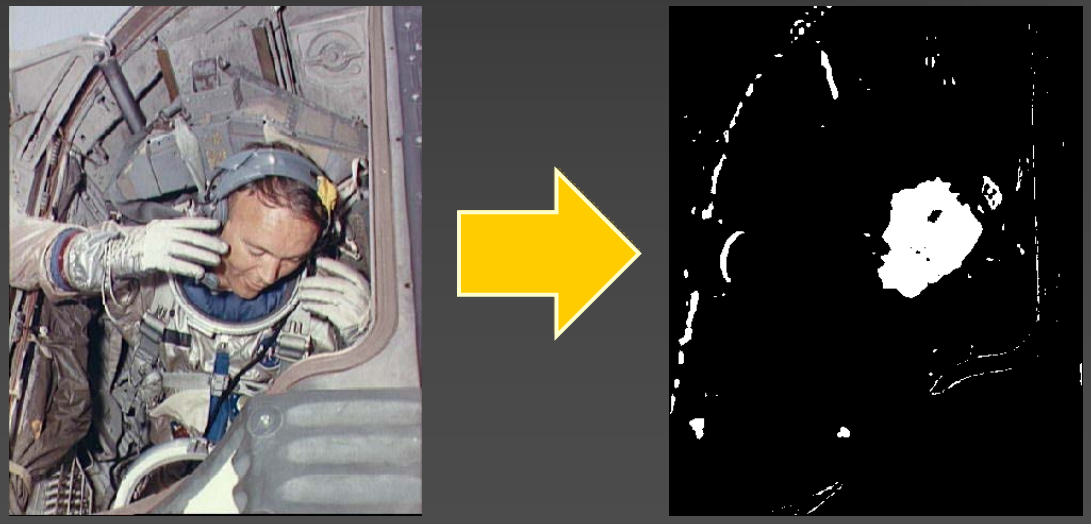
Step 1: YCbCr conversion

- Luma (brightness) stored in Y component
- Chroma (color) stored in Cb/Cr components
  - Allows for filtering the skin-color independent of brightness => simple rectangular region



Step 2: Skin-Color Filter

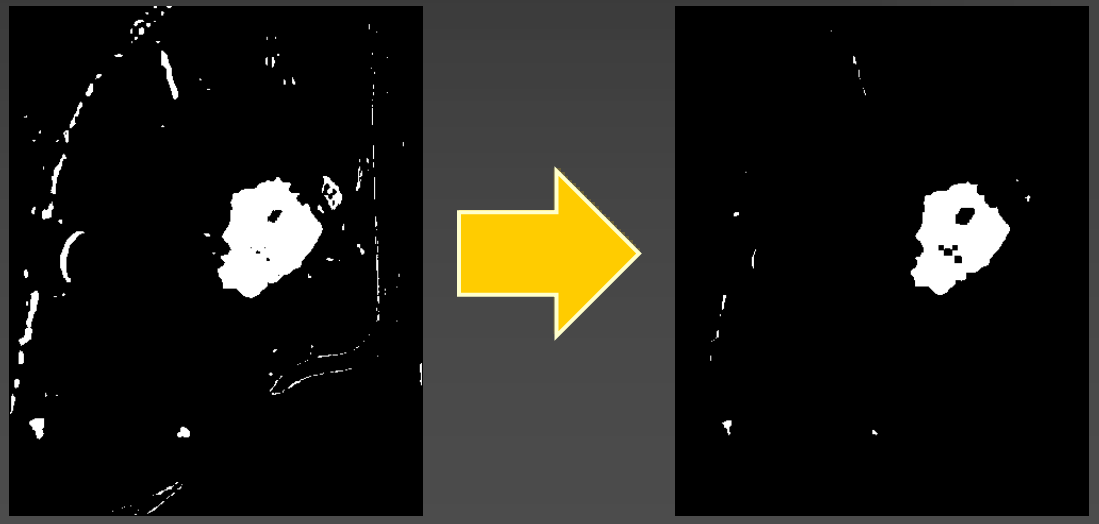
- Color value of every pixel is evaluated
  - Skin-colored pixels become white
  - Other pixels become black



NASA/courtesy of nasaimages.org

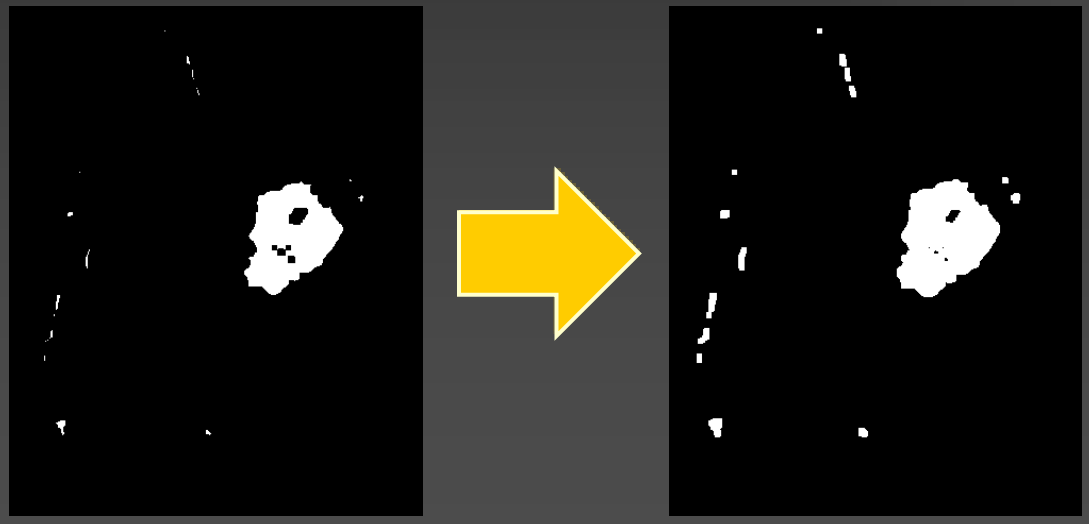
Step 3: Erosion

- Apply an erosion filter to remove noise
- White pixels, which have a black neighbor pixel, are blackened



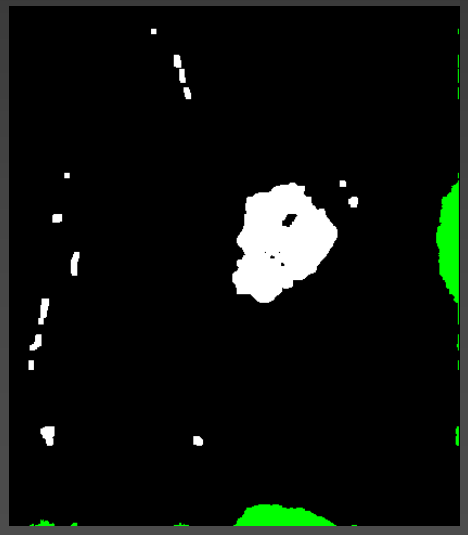
Step 4: Dilation

- Face size is reduced due to erosion
- => Dilation: opposite to erosion
  - Black pixels, which have a white neighbor pixel, are whitened



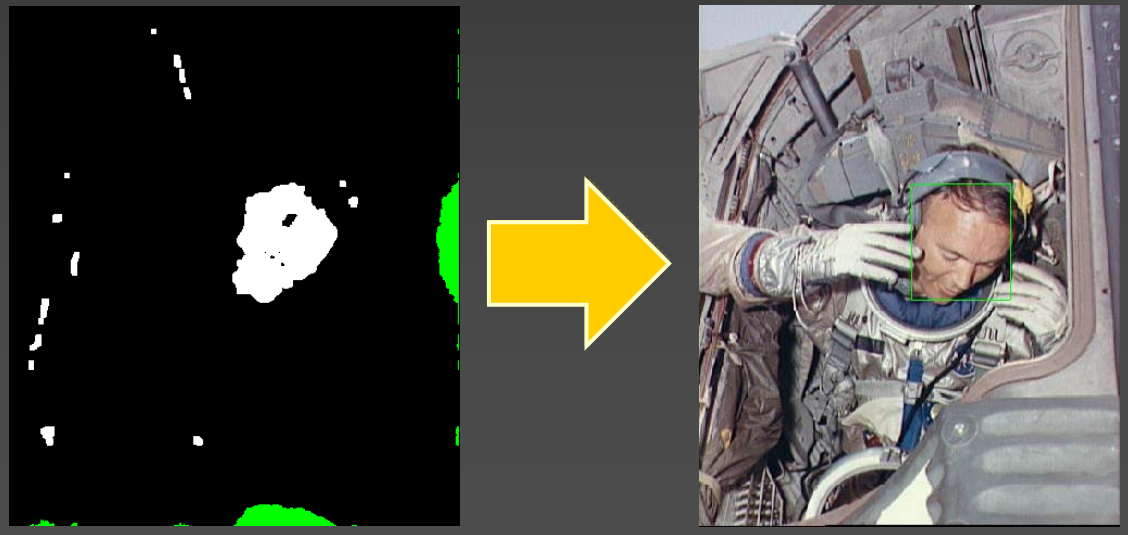
Step 5: Image Segmentation

- Find candidate face regions
- Histogram-based approach
  - Count white pixels in every row/column



Step 6: Compute coordinates

- Compute edges of each candidate region
  - Based on lower threshold in histogram
- Choose candidate region with biggest area
- Draw rectangle in original image



SW Development

- Provided application written in C
- Download over RS232 as SREC file
- Two application modes:
  - Normal mode
    - Image data is taken from camera
  - Test mode (make TEST=1)
    - A single image can be downloaded over the serial interface
    - Face detection is performed and the result image is uploaded to the PC
    - Execution time is evaluated

Some Hints

- Performance optimization by
  - Analyzing SW for repetative and time-consuming code sequences
  - Implementing HW-accelerators
    - Basic arithmetic functions
    - Complex computations
    - Try to find computations suitable for HW
    - E.g., large degree of parallelism
  - Pipelining computations

Some Hints

- Memory accesses might be bottleneck
  - Access of SDRAM by processor slow
    - Burst access not properly supported (high SW overhead)
  - Possible solution: Direct memory access of hardware modules on SDRAM
    - Module needs to be AMBA AHB master (compare svgactrl IP-Core from GRLIB)
  - Using fast SRAM memory could also help to migtigate this bottleneck
    - 2MB available

References

- Detailed explanation of face detection algorithm
  - http://channel9.msdn.com/coding4fun/articles/FaceLight--Silverlight-4-Real-Time-Face-Detection
- AMBA specification
  - http://polimage.polito.it/~lavagno/esd/IHI0011A_AMBA_SPEC.pdf
- Course Materials
  - http://ti.tuwien.ac.at/ecs/teaching/courses/hwswcode_lu