L14: Final Project Kickoff

Schedule - I

- Form project teams – by April 1st
- Project Abstract (Due April 6th in 39-553 by 1PM)
  - Start discussing project ideas with the 6.111 staff
  - Each group should meet with Prof. Akinwande to discuss the scope of the project and breakdown of tasks
  - Abstract should be about 1 page (clearly state the work partition) – a polished abstract will be published on the course website once your project has been finalized. Sample abstracts can be viewed at http://web.mit.edu/6.111/www/s2006/PROJECT/projects.html

    http://web.mit.edu/6.111/www/s2007/PROJECTS/projects.html
- Work in teams of two or three. A single person project requires special approval by the 6.111 staff.
- Proposal Conference with TAs (April 8-10). Bring your detailed proposal with you (3-5 pages including figures).
- Block diagram conferences with TAs (April 13-17)
  - Review the major components in the system and your overall design approach
  - Each group in discussion with TA, creates a deliverables checklist (i.e., what we can expect to be demonstrated) – (Due April 27th in class).
  - Specify the device components you need to acquire (*small* budget allocated for each project if component does not exist in the stock room). Get approval from the 6.111 staff and your TA will contact John Sweeney to obtain the parts.

Schedule - II

- Project Design Presentation (in 32-144) on April 22, 24 & 27
  - Each group will make an electronic presentation (power point or PDF)
  - Everyone is required to attend all days (not just the days you are presenting) – this will count in your participation grade. Each student will write comments (anonymous) which will be provided to the presenting group as feedback.
- Final project check-off (with teaching staff) on May 12. 30 minutes
- Final project presentations and video taping on May 13 (Wednesday): ~3-4 minutes (videos to be posted on the course website with permission)
- Final project report (in electronic format, which will be published with permission on the course website) due May 14 by 5:00PM (no late project check-offs or reports accepted)
- Kits have to be returned by May 14th at 5PM

Choosing A Topic

- You only have 5 weeks total (once your proposal abstract is turned in) to do this project. Be realistic in what you take on.
  - It is important to complete your project.
  - It is very difficult to receive an "A" in the class without completing the final project.
- The complexity should 2-3 times more than lab4 for each student. 12 units x 5 = 60 hours of effort.
- Quite often you will need to include analog building blocks (video, wireless, motors, etc.). However, keep in mind that this is a digital class and your design should demonstrate digital design principles.
- Complexity and innovation factor.
  - We will give credit to innovative applications, design approaches
  - More complex is not necessarily better

Grading - 36 points (%) Total

- Report and Presentation (6 points)
- Problem Definition and Relevance, Architecture, Design methodology (7 points)
  - What is the problem and why is it important
  - System architecture and partitioning
  - Design choices and principles used
  - Style of coding
  - All of the above should be stated in the project presentation and report
- Functionality (17 points)
  - Did you complete what your promised (i.e., graded by your customized checklist)
- Innovation (6 points)

Design Rules - 1

- Use hierarchical design
  - Partition your design into small subsystems that are easier to design and test
  - Design each sub-system so they can be tested individually
  - When appropriate, use Major/Minor FSMs
- Use the same clock edge for all edge-triggered flip-flops
  - Beware of clock skew
- Avoid problems from 'glitches'
  - Always assume that combinational logic glitches
  - Never clock a register from the output of combinational logic
  - Never drive a critical control signal such as write enable from the output of combinational logic
  - Ensure a stable combinational output before it is sampled by CLK.
  - Create glitch-free signals by
    - Registering outputs
    - Gating the clock

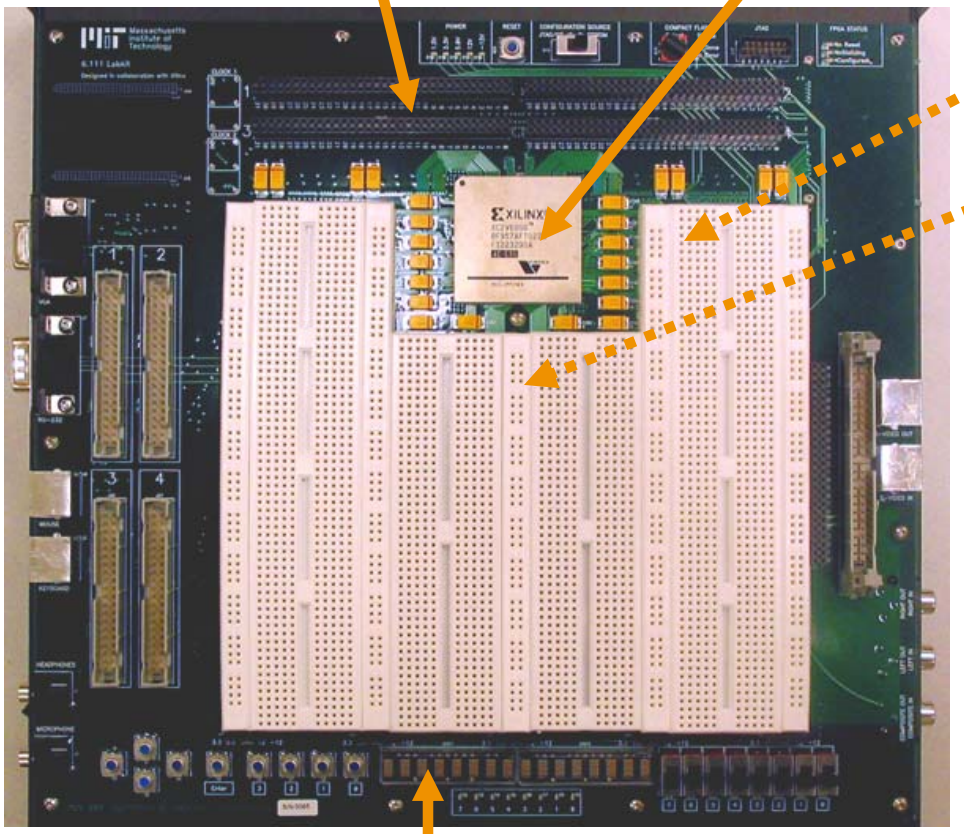
Design Rules - 2

- Avoid tristate bus contention by design
- Synchronize all asynchronous signals
  - Use two back to back registers
- Use memory properly
  - Avoid address changes when WE is true
  - Make sure your write pulse is glitch free
- Power supply can be noisy
  - Use bypass capacitors to filter noise
- Chip-to-chip communication
  - Beware of noise (inductance)
  - Might need to synchronize signals
  - Can also use “asynchronous” protocols

Debugging Hints - Suggestions

- Check your wiring!
- Read data sheet. Many controls signals use negative logic!
- Compile Verilog on the local drive.
- Backup Verilog files
- Stake out your labkit!

Labkit



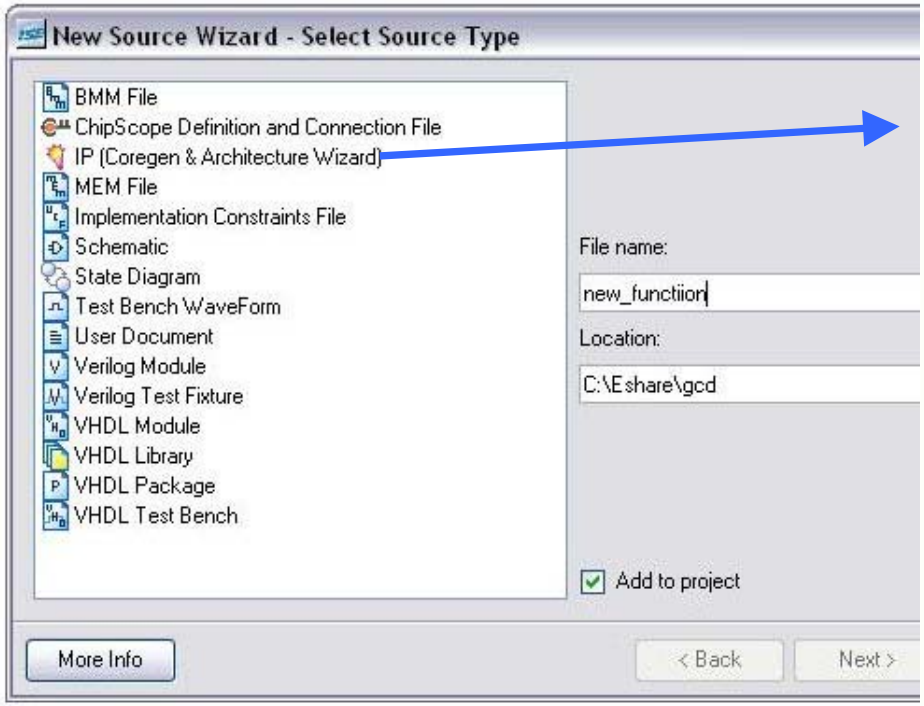
- Based on a huge Xilinx FPGA
- Built-in audio/video interfaces, high-speed memory
- Supports moderately high-speed designs (50-100MHz)

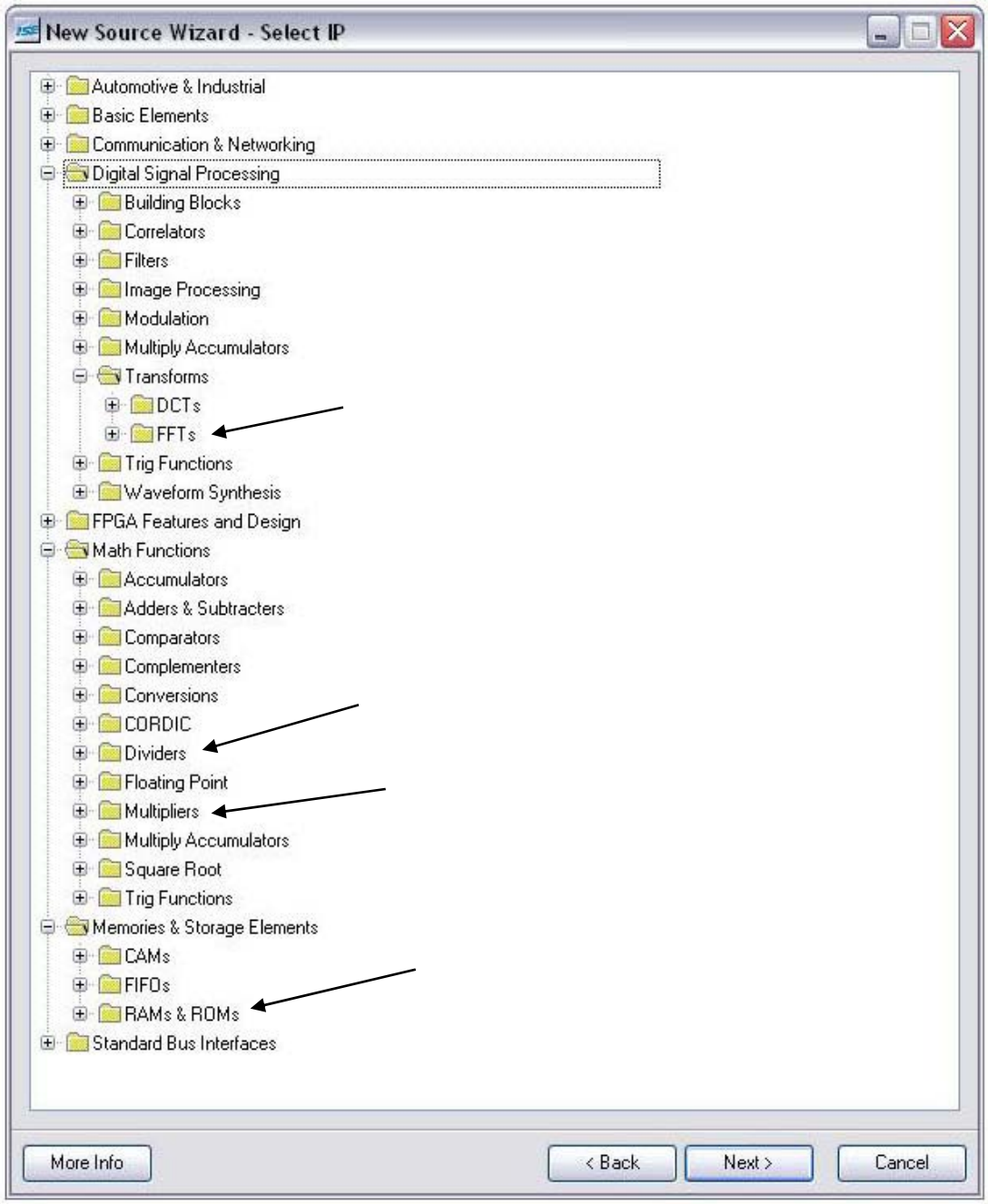
Important Odds and Ends

- Coregen
- NTSC (Camera Input)*
- Serial Communications*
- Motors

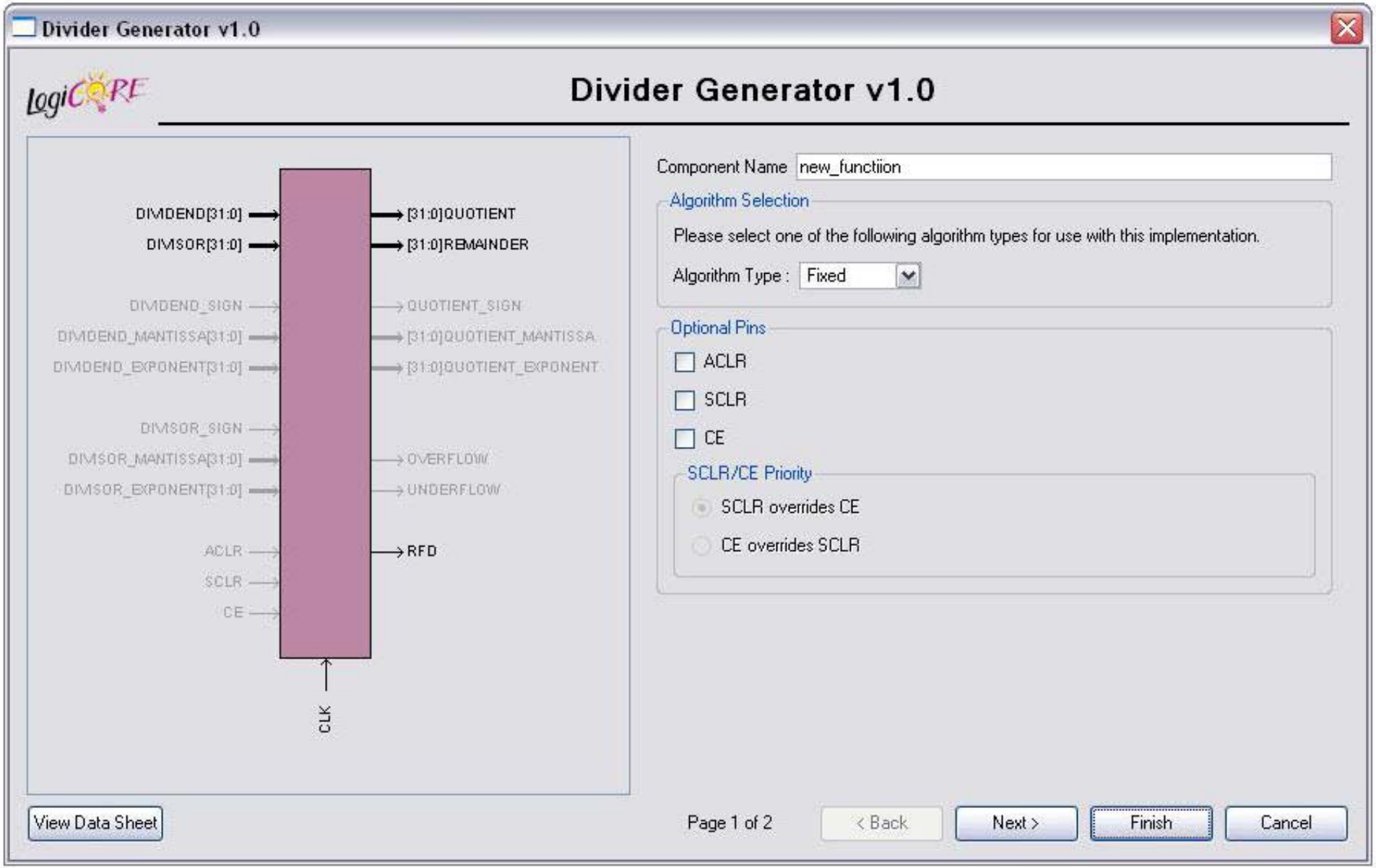
*Slides prepared by Dr Chris Terman

Coregen Wizard

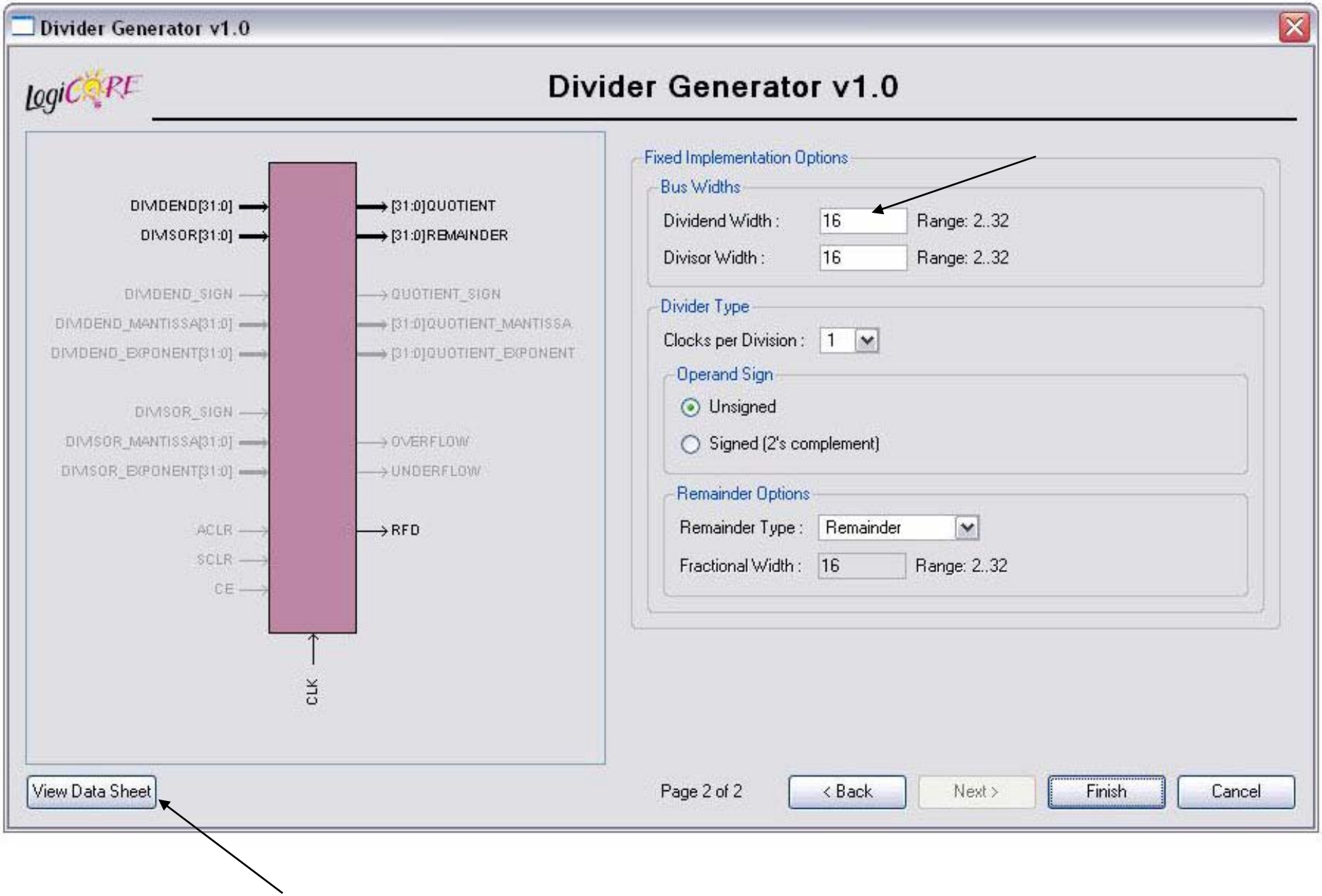




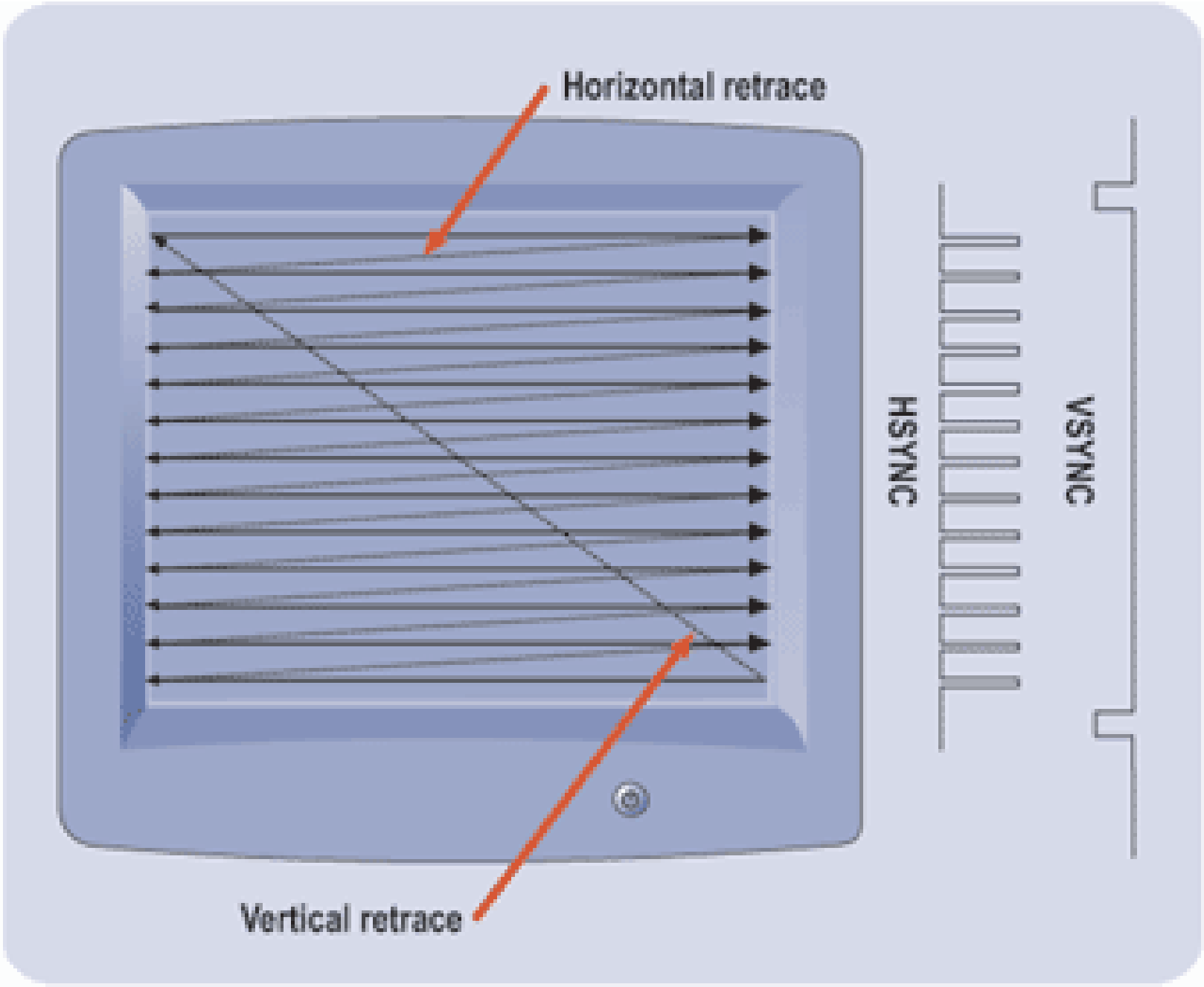
Coregen Divider



Coregen Divider



Sync Signals (HS and VS)



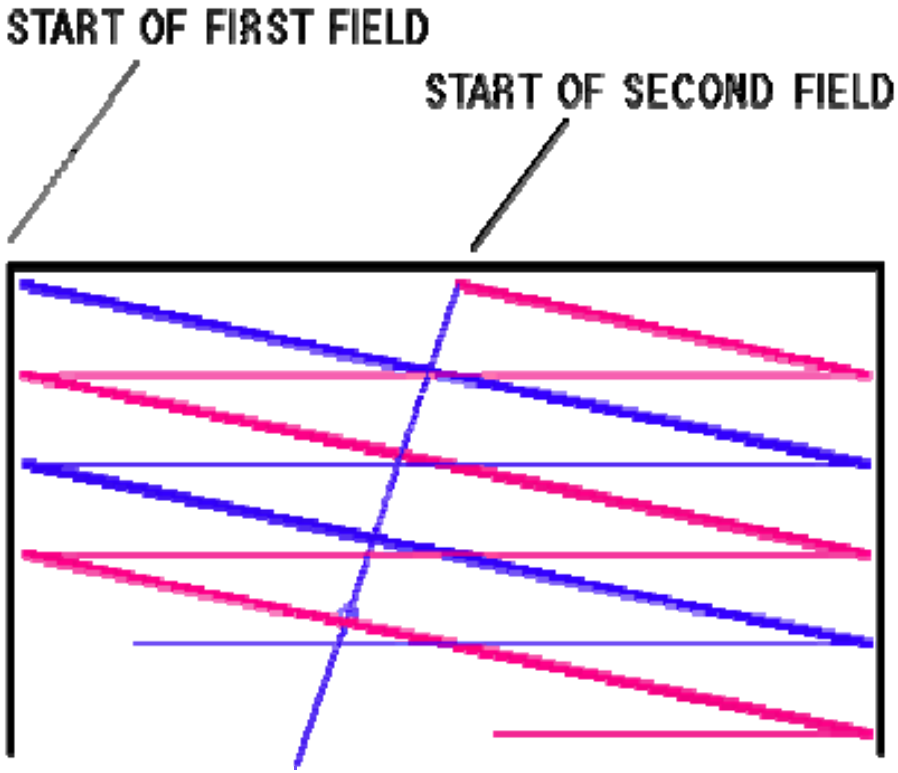
Interlace

Non-interlaced (aka progressive) scanning:

- VS period is a multiple of HS period
- Frame rate >= 60Hz to avoid flicker

Interlaced scanning:

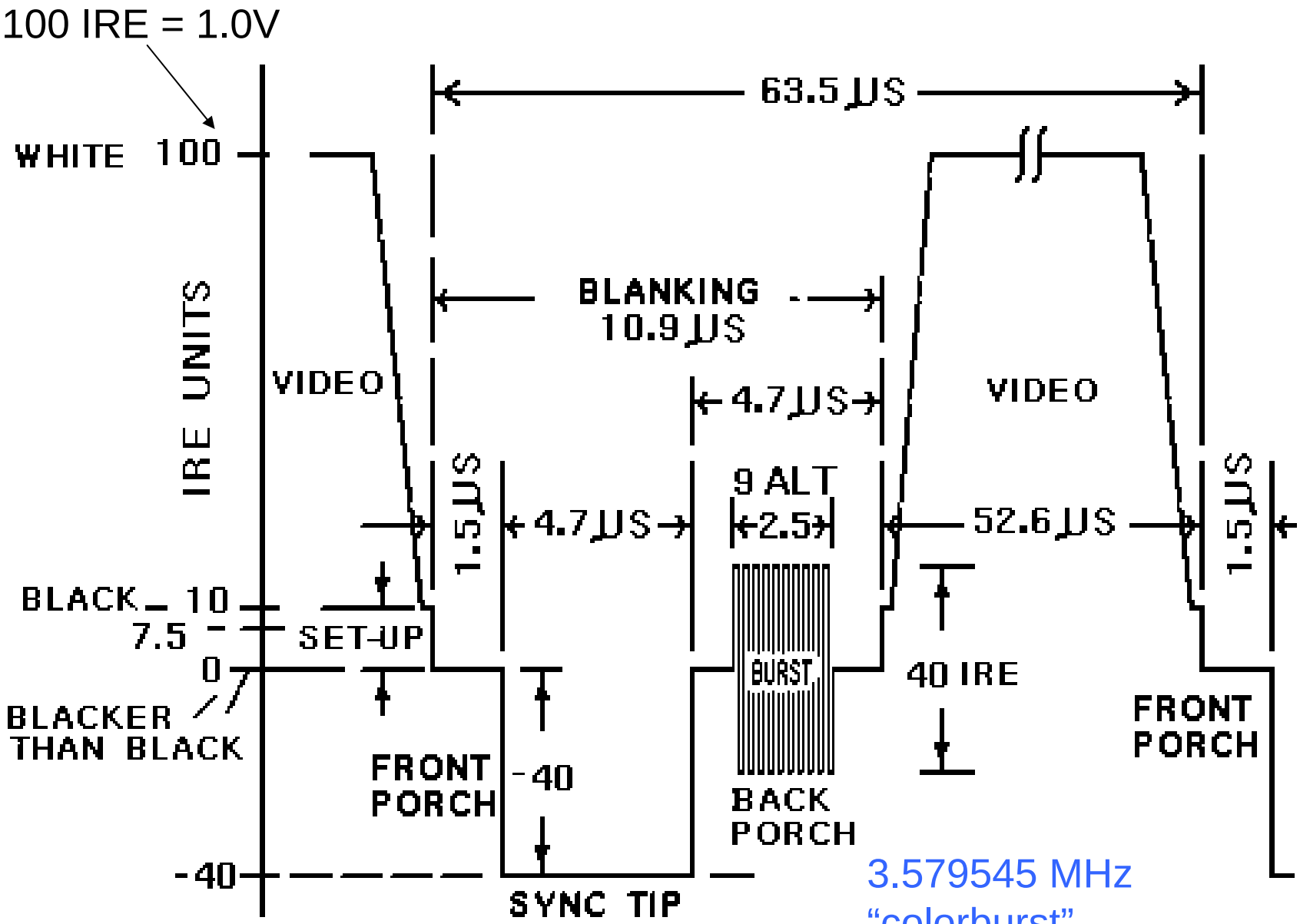
VS period is *not* a multiple of HS period, so successive vertical scan are offset relative to horizontal scan, so vertical position of scan lines varies from frame to frame.



NTSC example:

- 525 total scan lines (480 displayed)
- 2 fields of 262.5 scan lines (240 displayed). Field rate is 60Hz, frame rate = 30Hz

NTSC*: Composite Video Encoding



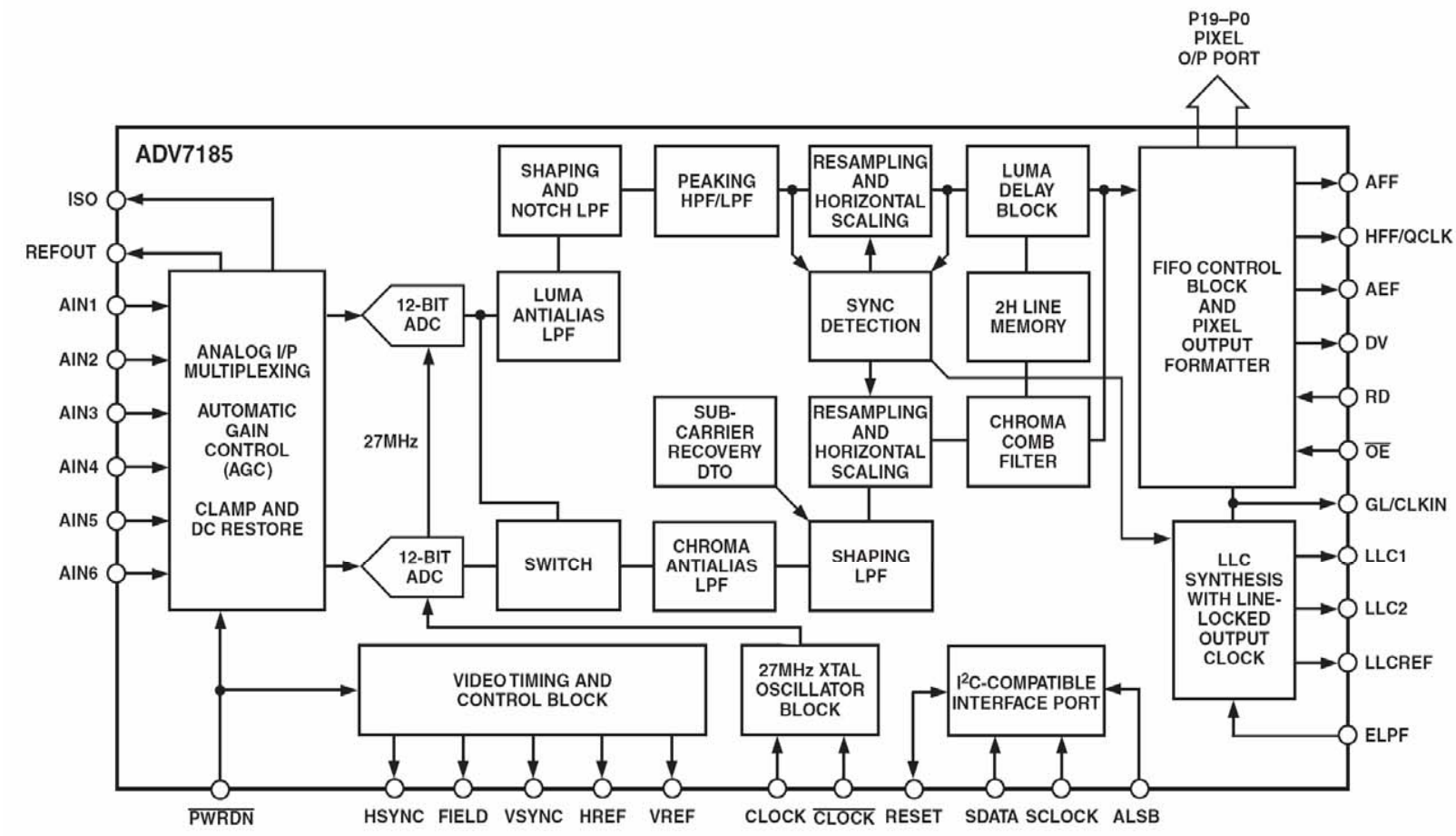
3.579545 MHz
"colorburst"

*National Television System Committee: 1940

Source: http://www.ntsc-tv.com

Labkit: ADV7185 NTSC Decoder

- Decodes NTSC and PAL video (composite or S-video)
- Produces CCIR656 (10-bit) or CCIR601 (8-bit) digital data



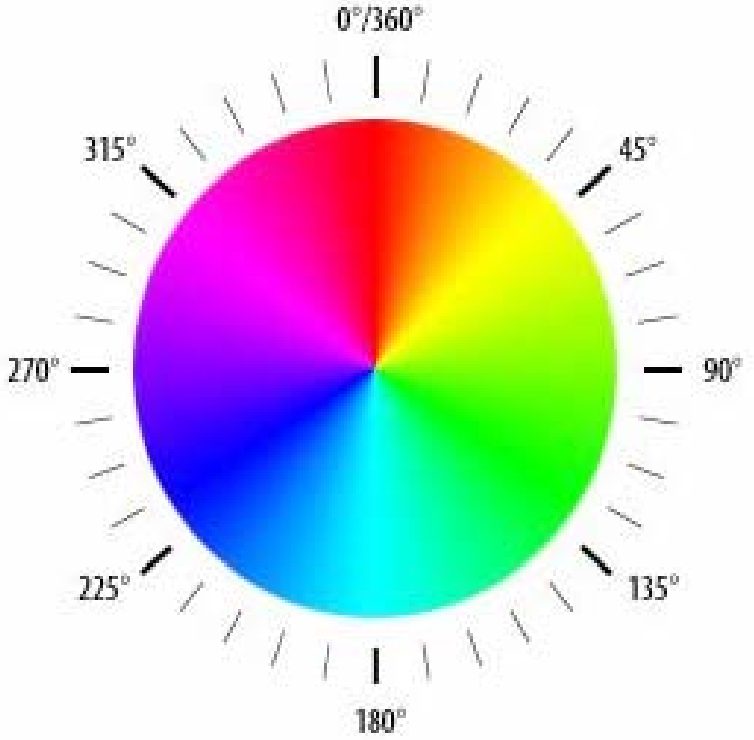
YCrCb to RGB (for display)

- Early TV broadcast in b/w (greyscale)
  Y = 0.299R + 0.587G + 0.114B
- 8-bit data
  - R = 1.164(Y – 16) + 1.596(Cr – 128)
  - G = 1.164(Y – 16) – 0.813(Cr – 128) – 0.392(Cb – 128)
  - B = 1.164(Y – 16) + 2.017(Cb – 128)
- 10-bit data
  - R = 1.164(Y – 64) + 1.596(Cr – 512)
  - G = 1.164(Y – 64) – 0.813(Cr – 512) – 0.392(Cb – 512)
  - B = 1.164(Y – 64) + 2.017(Cb – 512)
- Implement using
  - Integer arithmetic operators (scale constants/answer by 2^{11})
  - 5 BRAMs (1024x16) as lookup tables for multiplications

Video Feature Extraction

- A common technique for finding features in a real-time video stream is to locate the center-of-mass for pixels of a given color
  - Using RGB can be a pain since a color (eg, red) will be represented by a wide range of RGB values depending on the type and intensity of light used to illuminate the scene. Tedious and finicky calibration process required.
- Consider using a HSL/HSV color space
  - H = hue (see diagram)
  - S = saturation, the degree by which color differs from neutral gray (0% to 100%)
  - L = lightness, illumination of the color (0% to 100%)

- Filter pixels by hue!



Serial Communications

- Sending information one bit at a time vs. many bits in parallel
  - Serial: good for long distance (save on cable, pin and connector cost, easy synchronization). Requires “serializer” at sender, “deserializer” at receiver
  - Parallel: issues with clock skew, crosstalk, interconnect density, pin count. Used to dominate for short-distances (eg, between chips).
  - BUT modern preference is for parallel, but independent serial links (eg, PCI-Express) as a hedge against link failures.
- A zillion standards
  - Asynchronous (no explicit clock) vs. Synchronous (CLK line in addition to DATA line).
  - Recent trend to reduce signaling voltages: save power, reduce transition times
  - Control/low-bandwidth Interfaces: SPI, I^2C, 1-Wire, PS/2, AC97
  - Networking: RS232, Ethernet, T1, Sonet
  - Computer Peripherals: USB, FireWire, Fiber Channel, Infiniband, SATA, Serial Attached SCSI

RS-232 aka Serial Port

Characteristics

- Large voltages => special interface chips (Logic “1” = mark: -12V to -3V, Logic “0” = space: 3V to 12V)
- Separate xmit and rcv wires: full duplex
- Slow transmission rates (1 bit time = 1 baud); most interfaces support standardized baud rates: 1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K

RS232 interface

- **Transmit: easy, just build shift register to generate desired waveform with correct bit timing**
- **Receive:**
  - **Want to sample value in middle of each bit time**
  - **Oversample, eg, at 16x baud rate**
  - **Look for 1->0 transition at beginning of start bit**
  - **Count to 8 to sample start bit, then repeatedly count to 16 to sample other bits**
  - **Check format (start, data, parity, stop) before accepting data.**

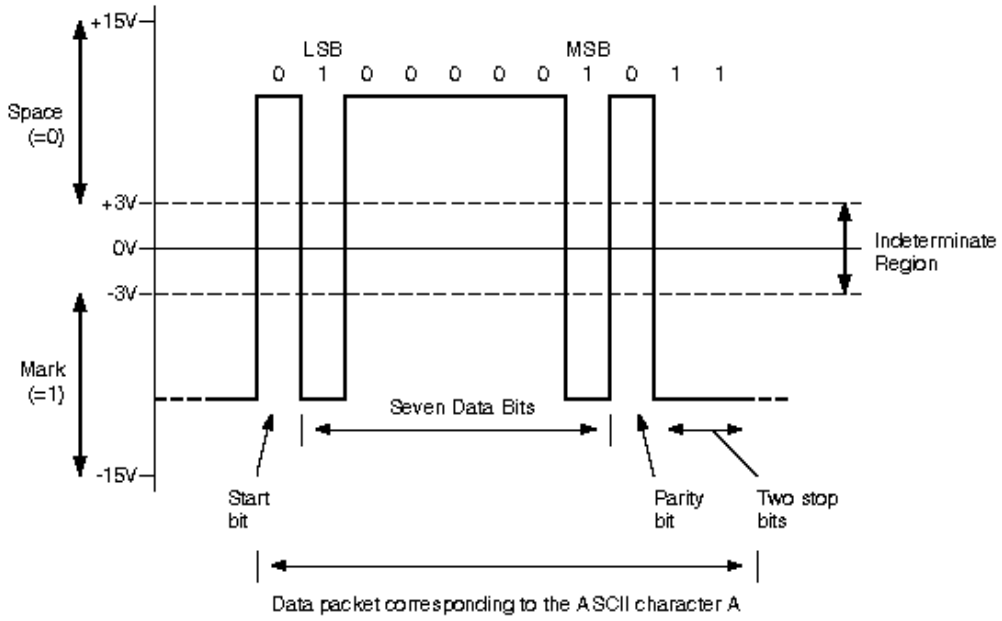
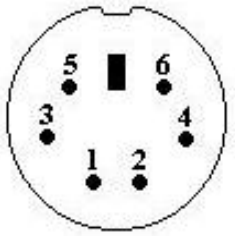


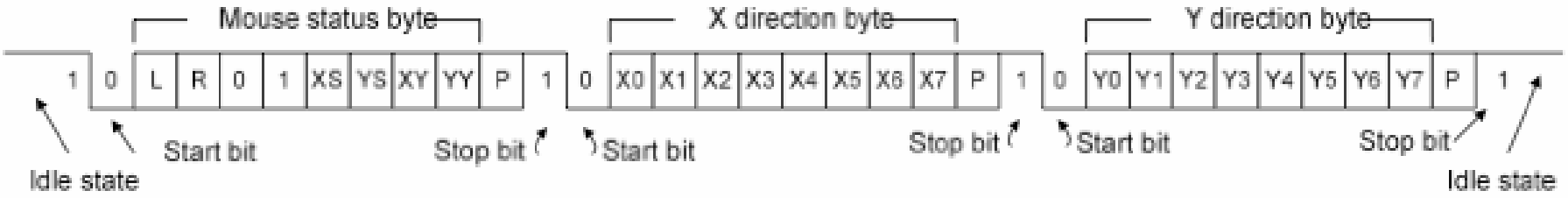
Figure from
http://www.arcelect.com/rs232.htm

PS/2 Keyboard/Mouse Interface

- 2 signal wire interface (CLK, DATA), bidirectional transmission of serial data at 10-16kHz



Pin	Signal	In/Out
1	Data	Out
2	N/C	
3	Ground	
4	+5V	
5	Clock	Out
6	N/C	



Figures from digilentinc.com

Stepper Motors

- DC motors with permanent motors and multiple coils around the body.
- Coils are turned on and off in sequence to cause the motor to turn.
- Because the coils are turned on and off they are easy to control with microcomputer and digital circuits. At any given time, the position of the shaft is known.
- Holding torque requires power.

Typical Stepper Motor

Bipolar Stepper Motor

- 2 phases
- 3.2 VDC
- 1.8° step angle
- Phase resistance: 4.8 Ohms
- Current: 800 mA
- Phase inductance: 3 mH
- Detent torque: 15 g-cm
- Holding torque: 50 g-cm
- Mounting hole space (diagonal): 1.02"
- Mounting holes: 0.1"
- Shaft diameter: 0.2"
- Shaft length: 0.83"
- Motor diameter: 1.38"
- Motor height: 1.02"
- Weight: 0.3 lbs.
- Ball bearing
- Dielectric strength: 500V, 50Hz/minute
- 18" lead wires
- Ambient temp.: -10°C to 55°C
- Insulation resistance: 100M ohms @ 500VDC
- Use P/N 161998 for shaft coupler

Jameco P/N	237420
Mfg	JAMECO RELIAPRO
Mfg #	35BYG002
RoHS?	No
In Stock	Yes

MOTOR,STEP,3.2VDC/.8A,
50 GM-CM

# of Units	$US EA
1+	12.15
10+	11.05
50+	10.05

Add to favorites

Order Quantity
1
Add to cart
View cart

View Catalog Page
Cat 261 , Page 149

Servos

- Servos are motors with electronic circuitry that controls the angular position of the shaft based on a control signal. If the angle is incorrect the motor is turned on until the correct position is reach.
- Angular position controlled by a 0 – 2.0 ms pulse width.

Plan Your Work!

- **Final project represents 5 weeks of work on a 12 unit course = 60 hours. You should average ~3 hours/day for four days /week - leaving you to enjoying a 3 day weekend!**