11/13,15 W13

11/20 W14 GPUs, Thanksgiving.

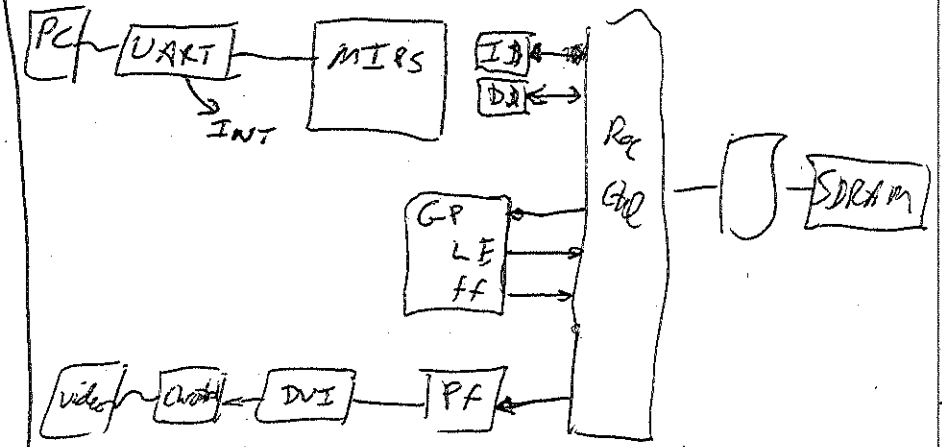
11/27,29 W15 Projects due

11/30 3PM Last checkoff oppty

12/3 RRR week prep.

12/14 Final Exam
↳ Questions at end of CP4 are fair game

use this to debug



- GP & PF both access SDRAM "directly" (not through processor)
- Known as DMA - direct memory access
- reality is that it isn't "direct" - need a DMA controller (Reg. Controller) to mediate

Why DMA?

Interfaces to I/O

when { polling, Interrupts, DMA

how { ports, memory map, DMA

iwio, swio, iw, sw } map to HW

swio "UART_TX, $t0

sw ~~$t0, UART_TX~~ $t0, 0($t1)
UART_TX

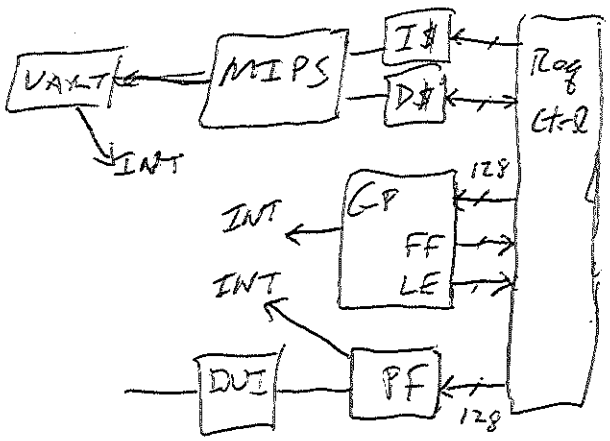
sw frame buffer

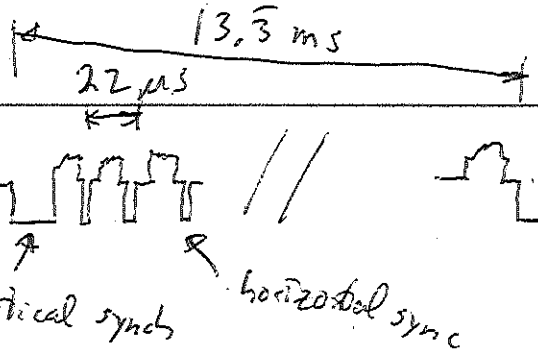
ports, e.g. x86

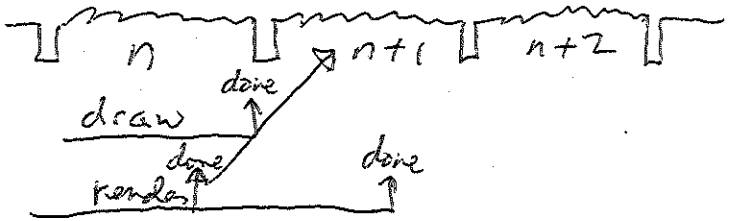
memory mapped not really in RAM

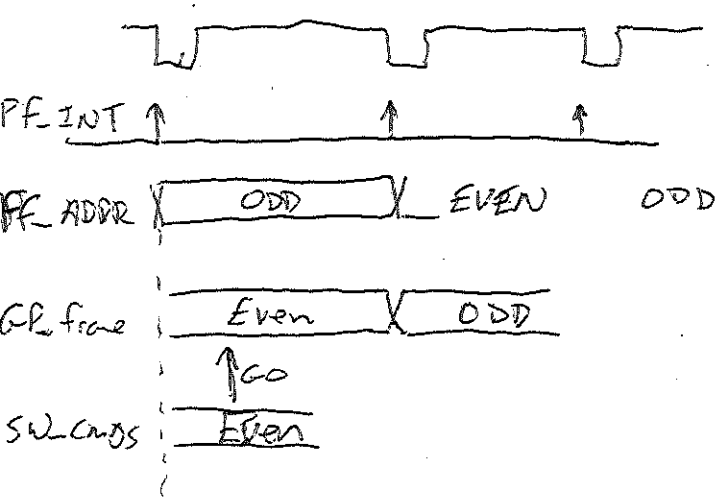
actually in RAM

Project:









write cmds, clear, renda, in same frame

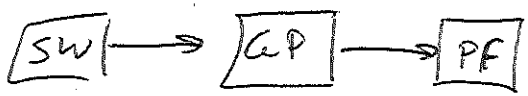
2 separate - more time
- more latency

Pixel feeder: ~~input~~ frame address (in)

GP: cmd list (in)

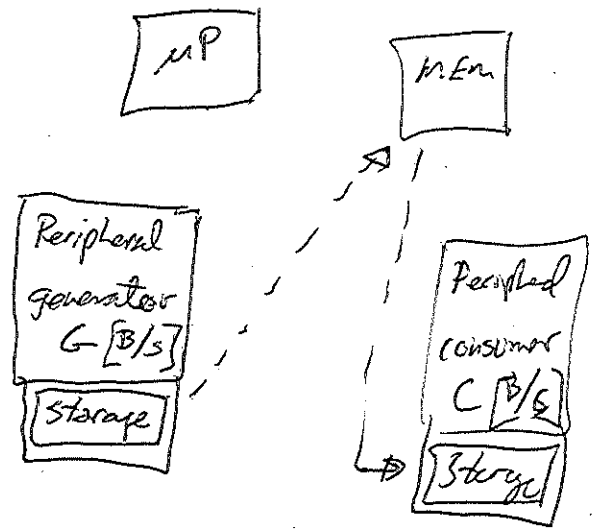
frame address (out)

SW: cmd list (out)



Often peripherals use a combination
- Memory mapped control registers
- DMA for data
- interrupts for notifications
- polling to figure out what happened

Examples
UART
Video
ethernet
radio
ADC
audio
DAC



Data is generally going to/from MEM
μP has an ISR latency T_{ISR}
peripheral has some storage S

if $G T_{ISR} > S$ lose input data

$C T_{ISR} > S$ lose output signal? BW?

need $\frac{S}{T_{ISR,max}} > G$ (or C)

So make S bigger or T_{ISR} smaller

115.2 KBps UART

$G = \sim 10$ KB/s

S = 1 B need $T_{ISR,max} < \frac{S}{G} = \frac{1\,B}{10^4\,B/s} = 10^{-4}$ s

not a problem for project.

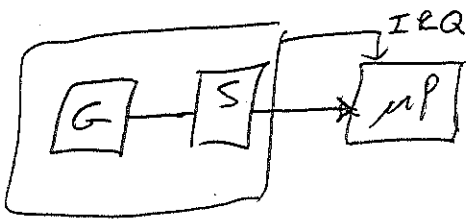
for PC? no way.

Video G = 1.5 MB/s

would S = 1 B work?

(FIFO)

So add more S inside peripheral



When do you generate IRQ?

1st byte? could be inneffiziend
last byte? back to S=1 case
S/2 ?

all make sense in different contexts

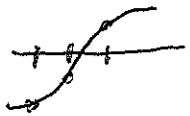
or use DMA

- no processor cycles needed for XFER (maybe cache miss delay)
- may need some FIFO space

$$S > G \cdot T_{\text{DMA latency, max}}$$

EX

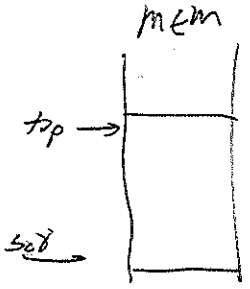
ADC, DAC
even at slow rate, need accurate sample interval



error in time is equivalent to error in value

need (Δt_{error}) [s] $\cdot$ (max slope) $\left[\frac{LSB}{s}\right] < LSB$

EX, cont



MMIO registers:

M_top, M_bot
rate
#samples
interrupt config

DMA for samples

GP & pf do DMA

DMA controller has many ports, each w/ a ready/valid interface, data, & addr.

af_addr_din af_cmd_din address FIFO 31 b

af_cmd_din 3 b

Writes

31 → af_addr_din

128 → wdf_din

16 → wdf_mask_din

need to do twice to push 256 bits

← af_full
← wdf_full
} ready

→ wdf_wr_en
→ af_wr_en
} valid

Reads

→ af_addr_din
→ af_cmd_din
← rdf_dout

twice to get 256 b

→ rdf_rd_en
← rdf_data_valid
← af_full
→ af_wr_en

GP commands

0x00, 000000 STOP

0x01 RRGGBB FILL

0x02 RRGGBB LINE
pt 1
pt 2 } top 2 bytes X, bottom Y for readability