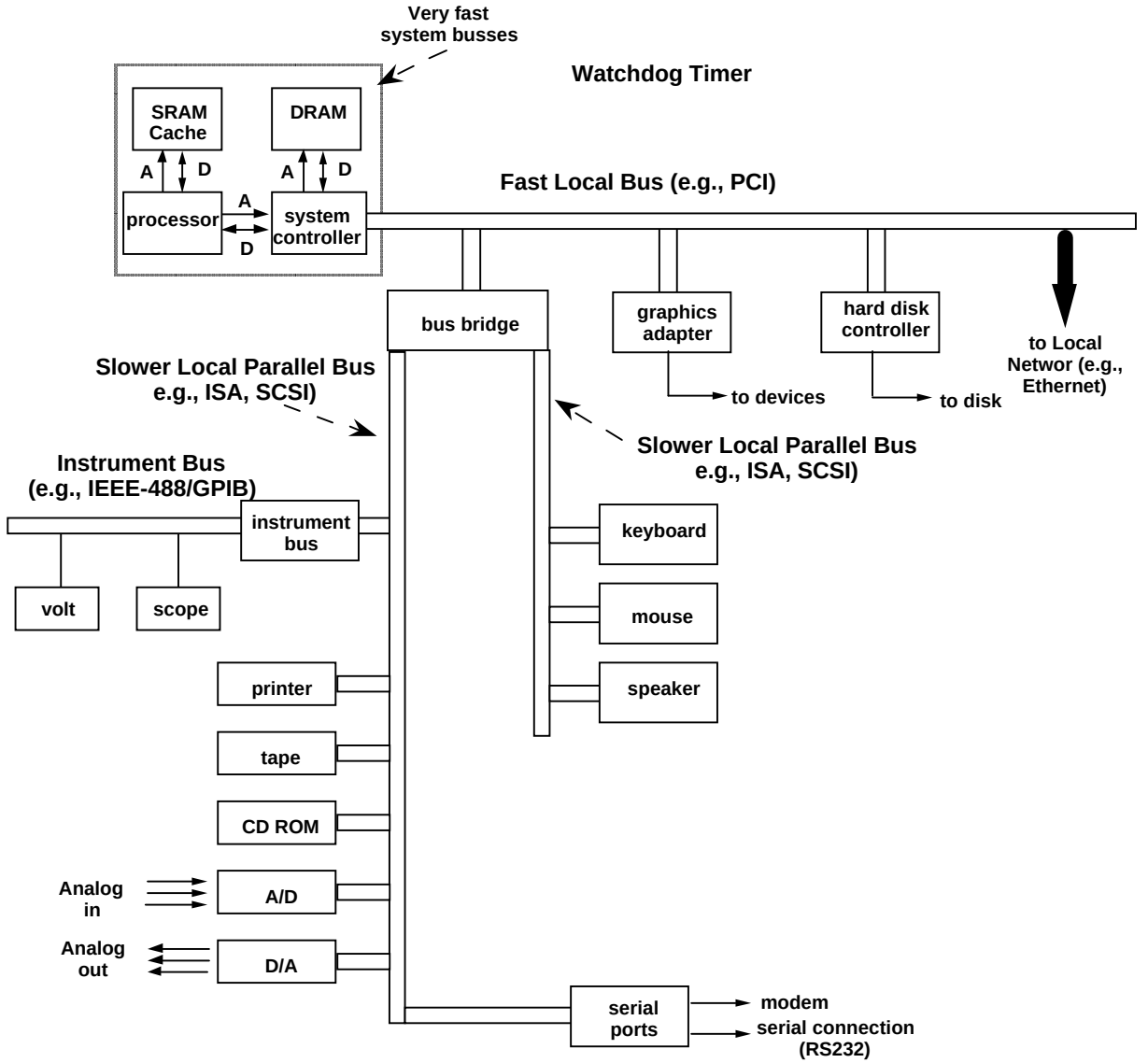
Communication, Flip-Flops, A/D, D/A
3/23/01 Lecture #19 16.070

- Introduction
  - Parallel and serial communication
  - Sequential logic → flip flops
  - Analog-to-digital and Digital-to-analog conversions

Typical Computer Bus Architecture

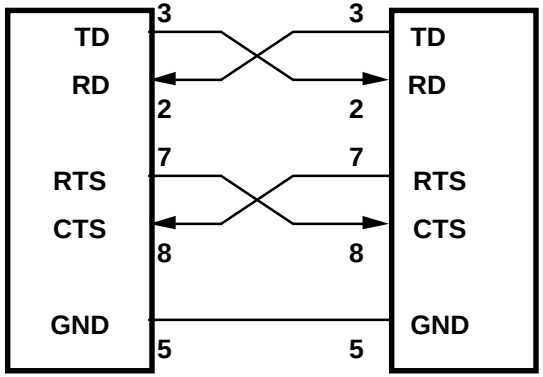


Notes on Computer Bus Architecture

- Very fast address (A) and data (D) system busses connect processor, system controller, static random access memory (SRAM) cache and local Dynamic Random Access Memory (DRAM). These system busses are processor specifics
- A fast local bus, controlled by the system controller, connects to very high data rate devices, such as graphics adapters and hard disk controllers
- A bus bridge controller hangs on the fast local bus, and controls slower busses
  - A slower parallel bus (e.g., ISA, SCSI) for high bandwidth peripherals
  - A serial link for medium/low bandwidth peripherals
- Sometimes the function of the local fast and slower busses are combined

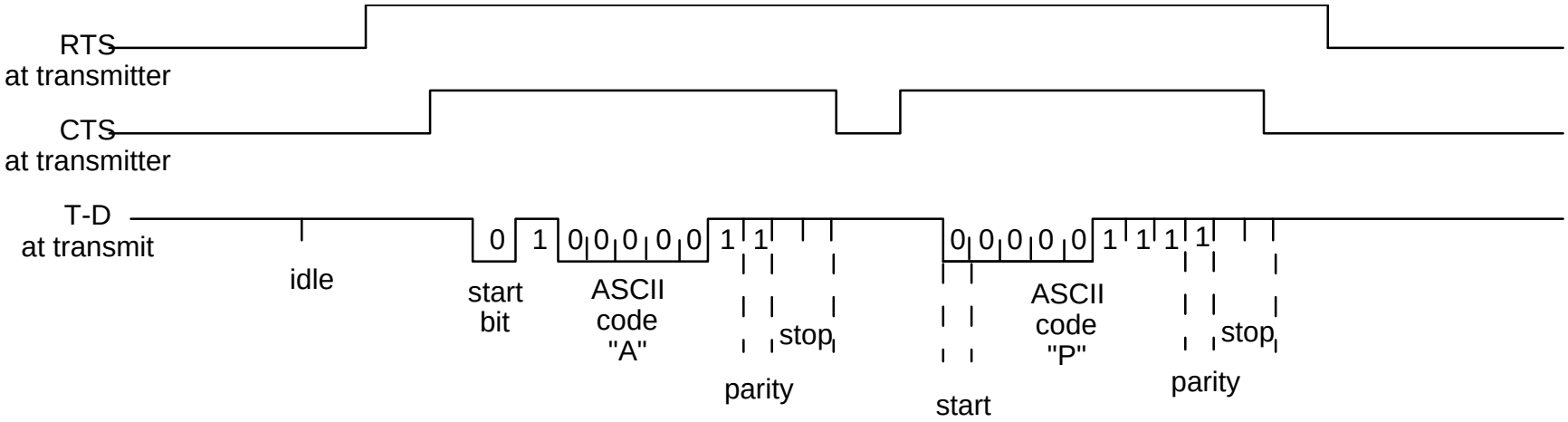
Serial Connections

- Typically have 1 or 2 data lines, a few control lines, and ground
- Data are sent in a serial stream of bits down the data line
- Oldest and most common is RS-232
  - TD - transmit data: data are sent via this line
  - RD - receive data: data are received here
  - RTS - request to send: set when unit wants to transmit
  - CTS - clear to send: set when ready to receive



- Standard RS-232 is 19.6K BPS (~1960 bytes/sec)

RS-232 Timing and Framing



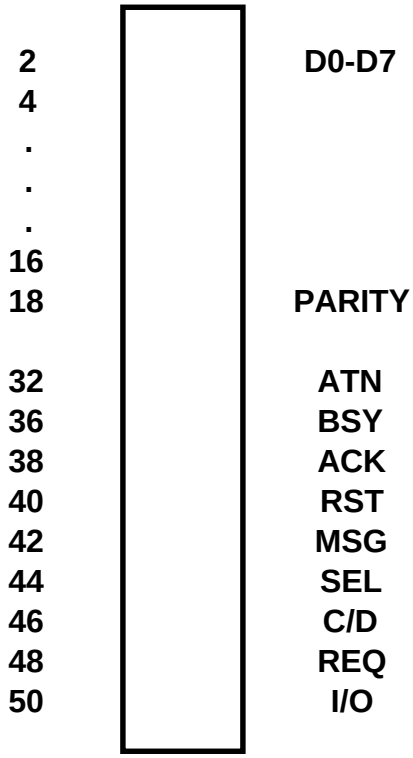
- RTS goes high, ready to transmit
- CTS goes high, go ahead
- Idle (1) drops to 0, a start bit to start the frame. Then the 5, 6, or 7 data bits, LSB → MSB, (here ASCII code), then 0 or 1 parity bits, then 1, 1.5, 2 stop bits (in duration)
- CTS goes low, as receiver clears buffer (it may or may not depending on how fast the receiver is)
- Second data frame starts

Parallel Busses

- All parallel busses have in common
  - A parallel set of lines for data
  - A parallel set of lines for address (which in more sophisticated busses may be multiplexed with data)
  - Command lines to indicate when reads and writes should occur, and to signal willingness/not to receive
- Parallel busses differ in
  - Mechanical arrangement of pins
  - Number of devices per bus
  - Control by controller/initiator
  - Width (i.e., how many lines)
  - Speed/frequency
  - Parity Checking

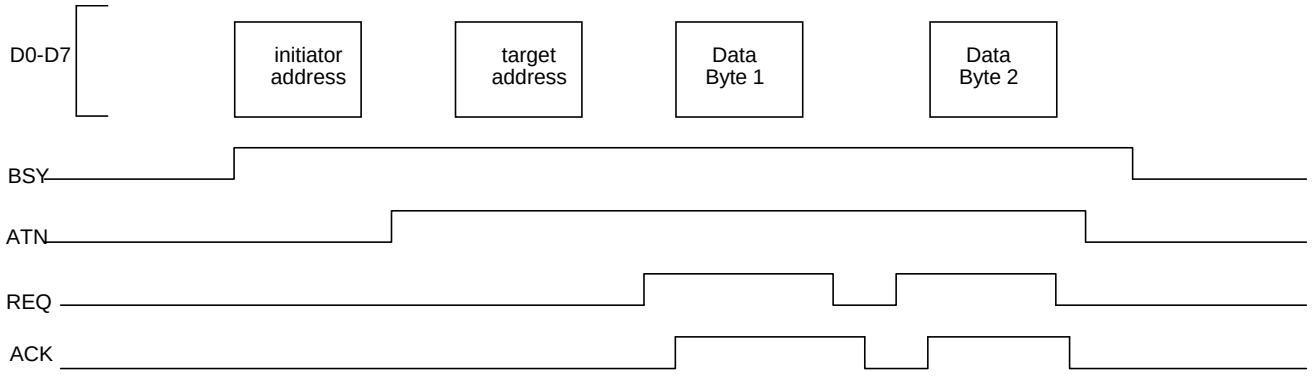
Parallel Bus - SCSI

- SCSI (Small Computer System Interface) is a daisy-chain parallel bus for up to 7 (or 15) high bandwidth peripherals
- Each device has a SCSI address 0-7 of which one is the "host" adapter
- No device is always the controller, but at any instant, one device is the initiator, a second the target, and only these two communicate at any one time
- The initiator takes control of a quiet bus and requests an action of a target
- Communication at 10-20MBPS



SCSI Timing and Sequencing

- A device takes control of the bus by raising BSY (busy) and its address on D0-D7. This lets all others know it's the initiator (if two try at once, there is arbitration that uses SEL (select))
- The initiator then notifies the target by raising ATN (attention) and the address of the target on D0-D7
- The target raises REQ (request) saying it is ready
- The initiator puts the first byte on D0-D7 and raises ACK (acknowledge)
- The target reads D0-D7 and drops REQ
- The initiator drops ACK

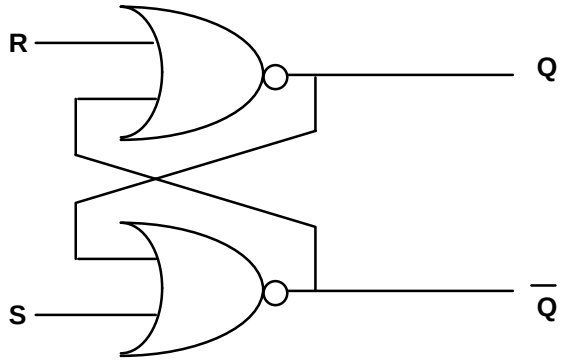


A Third Option - Packetized Data

- Parallel busses connect many devices, and use many parallel lines for address, data, and commanding
- Serial connections connect two devices, and use 1 or 2 lines for data, and a few for commanding
- Packetized data uses only 2 lines (like a serial connection) but can connect many devices
  - Transmitting device creates a serial packet which has a header, which includes its address, and the address of the target
  - Transmitting device waits for quiet line, and puts its packet, serially, on the common line
  - All devices listen for their address in header, copy their packet and (sometimes) acknowledge sender
- Various levels of implementation
  - USB (Universal serial bus) to mouse, keyboard, etc.
  - Ethernet in LAN (Local Area Network) in an office
  - TCP/IP (Transmission Control Protocol/Internet Protocol) in the internet

Sequential Logic

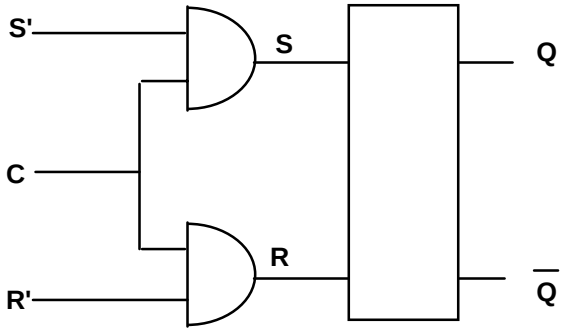
- Up until now, all logic has been combinational - the output is determined completely by the existing state of the inputs.
- Sequential Logic creates memory in a set of gates
- Simplest is an S-R flip-flop
  - S and R Lo → no change
    - Q = 0, $\overline{Q}$ = 1 and Q = 1, $\overline{Q}$ = 0 both allowed
  - S Hi, the output (Q) is Set to 1, regardless of previous state
  - R Hi, the output (Q) is Reset to zero



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	indeterm.

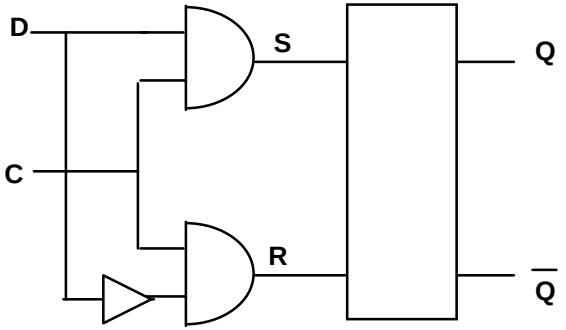
Clocked Flip-Flops

- Clocked R-S
  - Performs RS function when clock strobes



S'	R'	Qn+1
0	0	Qn
0	1	0
1	0	0
1	1	--

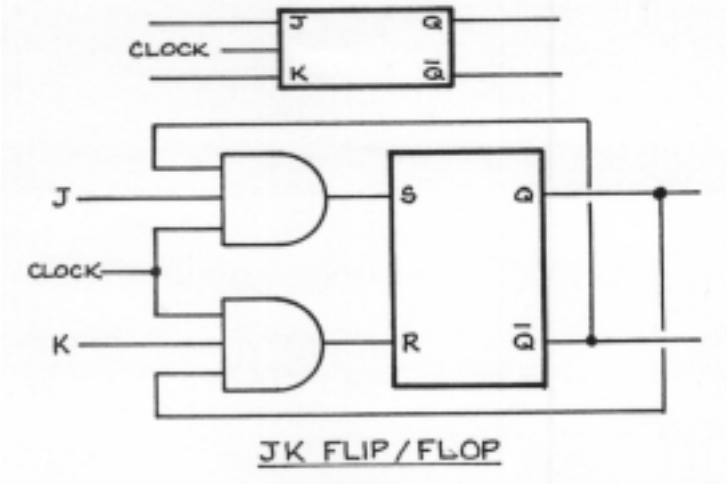
- D Flip-Flop
  - Latches Data line
  - The element of Latch in Static Ram



D	Qn+1
0	0
1	1

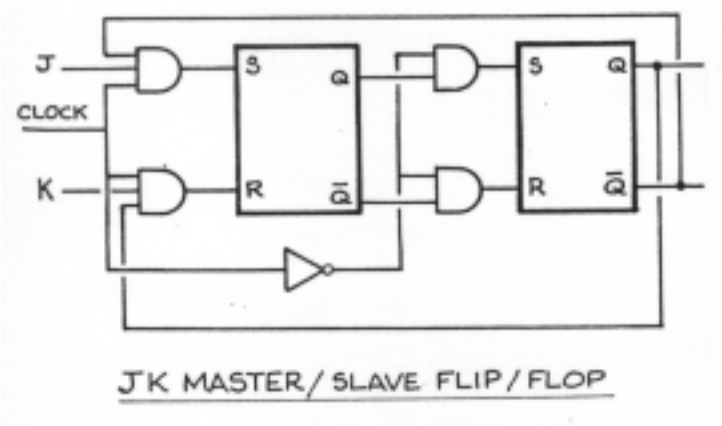
J-K Flip-Flops

- J-K Flip-Flop
  - Same as S-R except J=1, K=1 "toggles" output



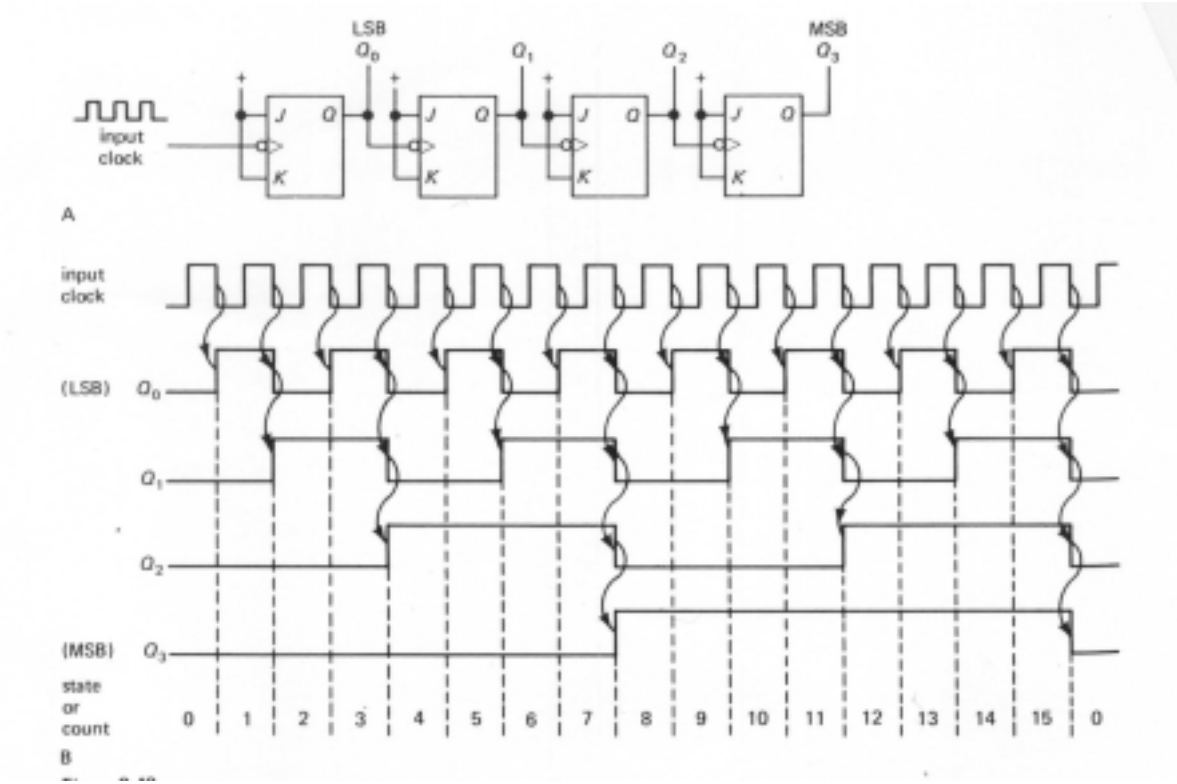
J	K	Qn+1
0	0	Qn
0	1	0
1	0	0
1	1	$\overline{Qn}$

- Master-Slave J-K
  - Same as J-K except master flips when clock rises, slave flips when clock drops



A Flip-Flop Counter

- Output counts from 0 to $(15)_{10}$ and then resets to zero

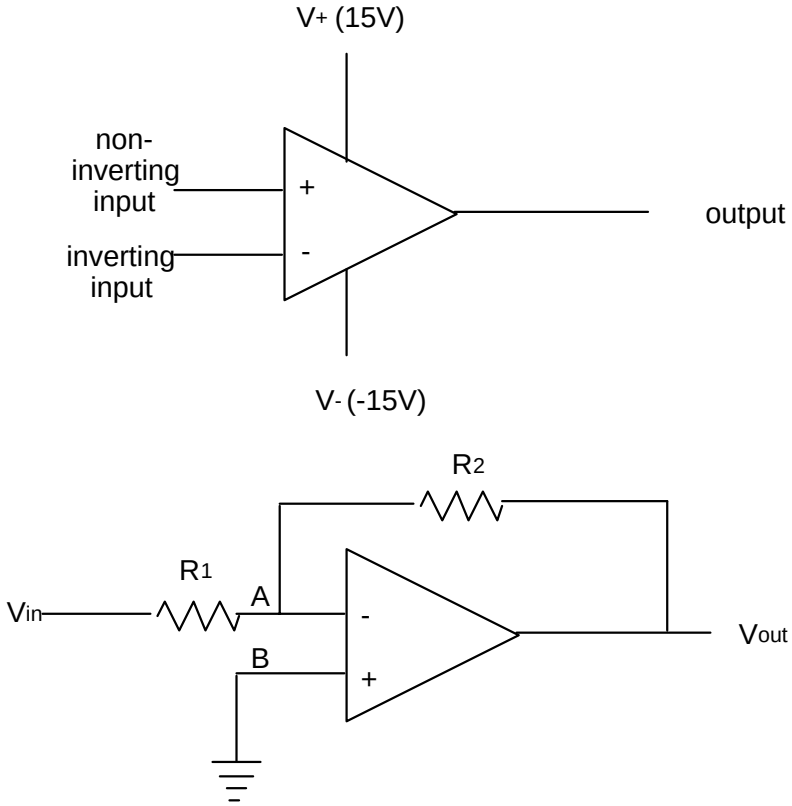


Analog to Digital Conversion

- Often analog signals must be converted to digital information to be read into computer - the A/D converter or ADC (invented by Bernie Gordon '48)
- Function is as follows:
  - Upon receipt of start command, clock is sent to digital counter where it increments
  - Digital signal is converted to analog ramp
  - Ramp is compared with analog in, and when they cross, comparitor signals
  - Clock signal to counter is stopped, last count is latched and data ready flag set
- About 6 other successful methods of ADC exist, so choice of which depends on speed, accuracy and cost requirements of application

Aside - Op Amps

- Operational amplifiers are high gain (10^5 to 10^6), differential, single output amplifiers, which with common external feedback resistances, are stable (a la Steve Hall). They are the building blocks of most analog circuits
- Without feedback, any difference at the inputs will drive the output to the voltage rails

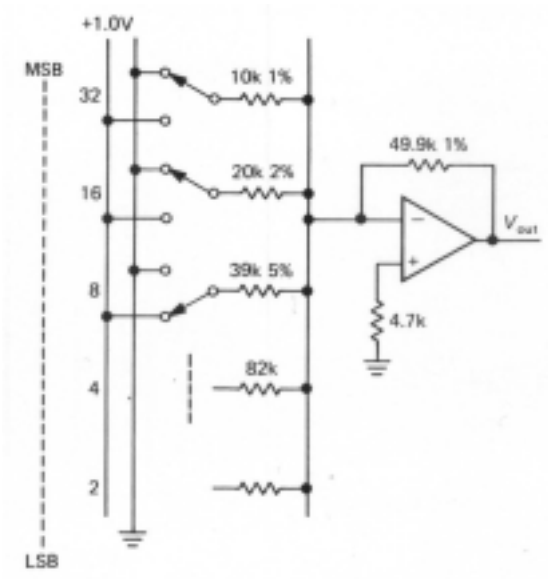


Aside - Op Amps

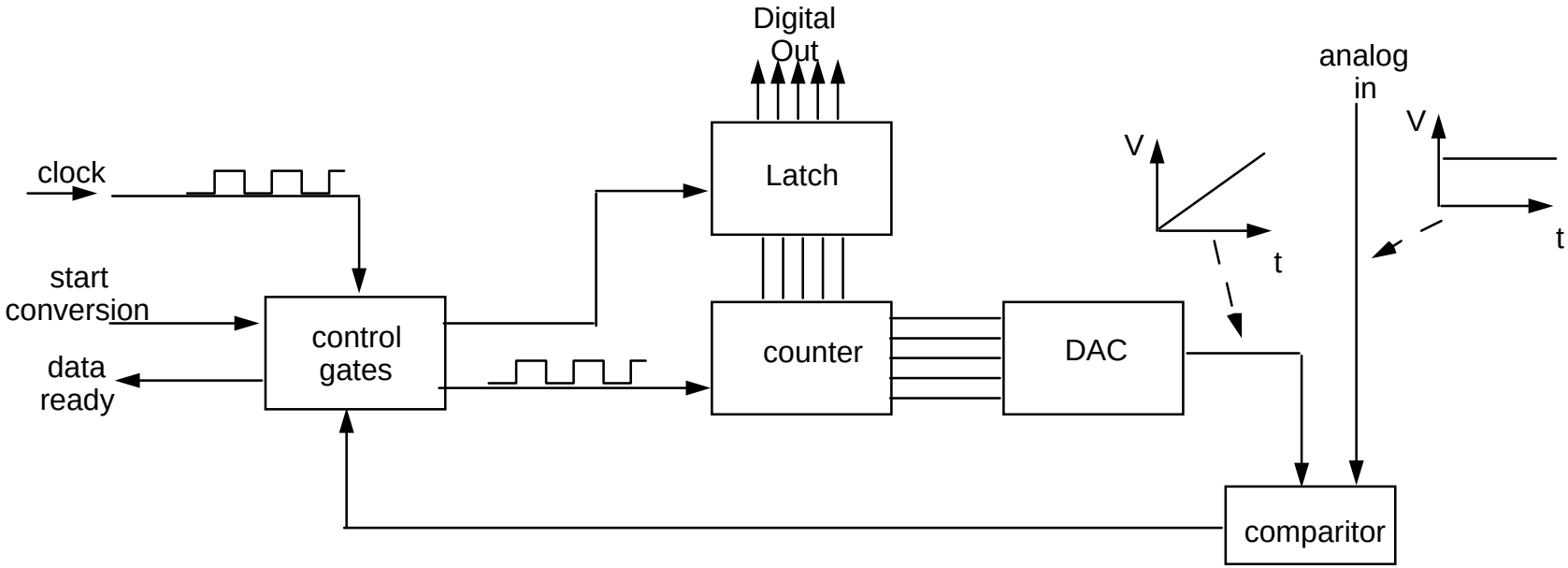
- Golden rules of op amps with external feedback
  1. The gain is so high, the output attempts to do whatever is necessary to make the voltage difference between the inputs zero
  2. The inputs draw no current
- So for the simple circuit:
  - Pout B is at ground, so by rule 1, so is A
  - So the voltage across R_2 is V_{out}, and across R_1 is V_{in}
  - So by rule 2, since they have the same current: $V_{out}/R_2 = -V_{in}/R_1$
  - OR: $V_{out}/V_{in} = -R_2/R_1$

Digital to Analog Conversion

- Have digital number - want corresponding analog signal → DAC
- Simplest DAC uses a ladder of resistors and an operational amplifier acting as a summing junction
- Switches are actually TTL or FET's

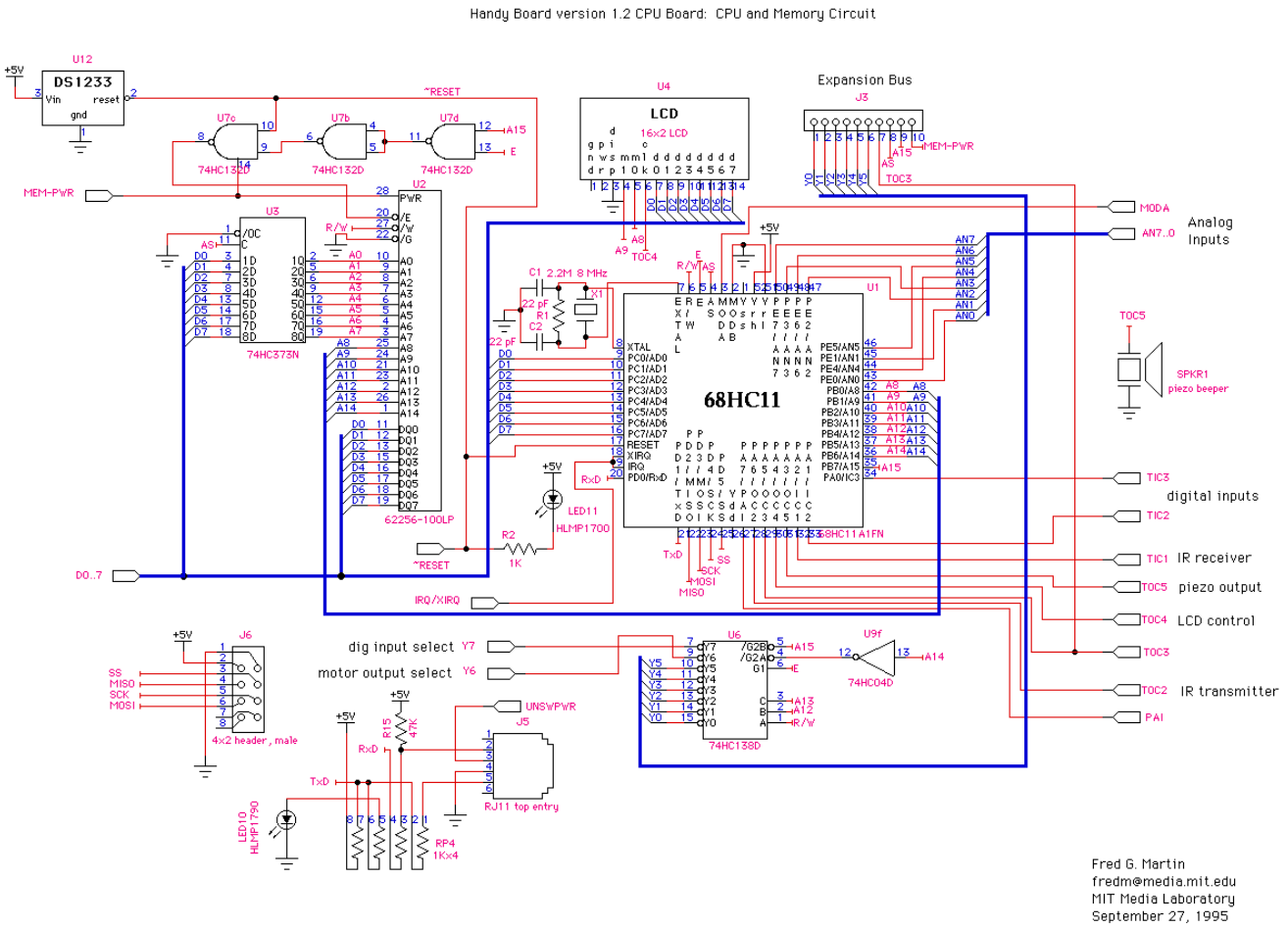


ADC Conceptual Circuit

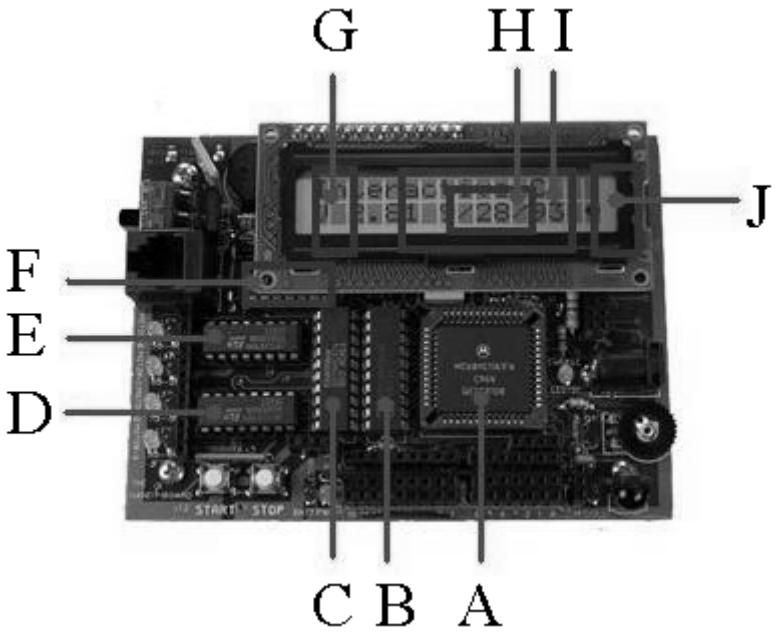


How would you implement these elements?

Handyboard -- What Have We Learned?



Handyboard Index



A. **68HC11A Micro-Processor**: Motorola micro-processor with address bus, data bus, 8-bit A/D converter. One A/D channel used for thumb wheel position conversion, IR Tx circuit connected directly to chip.
B. **74HC244 Tri-state buffer:** External digital inputs are buffered by this IC, before provided to processor. Tri-state allows for disconnection from processor data bus when no digital input received.
C. **74HC374 Octal Latch:** 8 D-type latches that store the motor controller settings provided by the processor, so that the processor data bus my me utilized for other tasks. (i.e. just keeps motor controllers doing what they are doing until the processor issues a different setting)
D. **L293D Motor controller (or equiv.)**: PWM motor controllers capable of output voltage swing for forward/backward rotation motor control. Two LEDs connected to outputs from chip to each motor are connected to the PWM backw./forw. outputs, indicating direction of rotation. Two motor controllers per L293D IC.
E. **L293D Motor controller (or equiv.)**: (See D.)
F. **74HC04 Hex Inverter:** Inverts six digital input bits. Four bits used between motor controllers and D-latch (See C.), to satisfy controller requirements. One bit used to address expansion bus J3. One bit used by IR Tx circuit.
G. **74HC132 Quad NAND gates:** Four NAND gates. Three used for reset/start-up sequence generation for memory (See I.). One gate used by IR Tx circuit.
H. **74HC373 Octal Latch (transp.):** Acts as address bus expander for processor. Expands 8-Bit address bus to 14-Bit by latching partial addresses provided by processor's data bus.
I. **62256-100/120LP 32K Static CMOS RAM:** 32K static CMOS system RAM.
J. **74HC138 3 to 8 Decoder:** Provides device address decoding for a maximum eight devices on the expansion bus J3.

Summary

- You have learned a lot about the building blocks of modern electronic computers
  - The basic building block
    - Transistors form gates, which form logic and amplifiers
  - The elements
    - Combinational logic (adders, control)
    - Sequential logic (latches, memory)
    - Op amps to handle analog interfaces
  - Some important assemblies
    - Address, SRAM, Busses, ADC, DAC
- The missing element is what goes on in the CPU

- Try utilizing this knowledge to decipher real machines -- goal is to understand and demystify (analyzing and designing takes much more knowledge)

References

1. P. Horowitz and W. Hill, "The Art of Electronics," Cambridge University Press, 1980.

2. W. Buchanan, "Computer Busses: Design and Application," CRC Press, 2000.

3. D. Phillips, "Modern Electronics and Communications," Marine Publishing, 1986.

4. R. White, "How Computers Work," QUE Publishing, 1999.