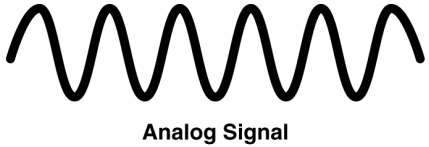
# Digital Signal Processing

Samantha Young, Lauren Pudvan, Vivien Chen

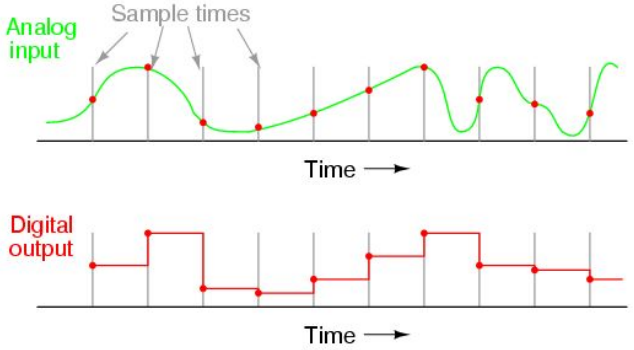
# What is DSP?

# Digital Signal Processing

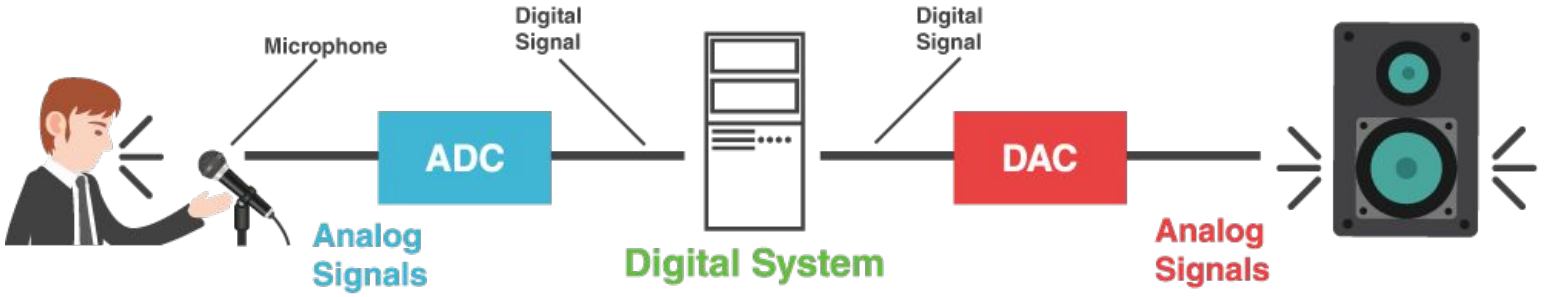
- Digital signal processing is the processing of digitized discrete-time sampled signals.
  - takes analog signals like audio, voice, video, that have already been digitized and then manipulates them mathematically.
  - Math can be done by general-purpose computers or by digital circuits such as ASICs or on digital signal processors.

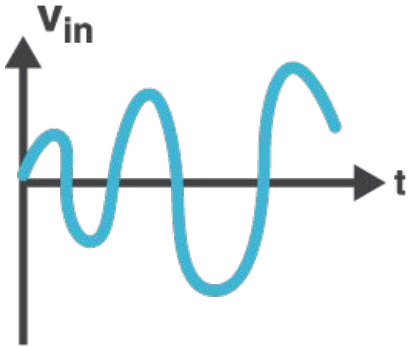


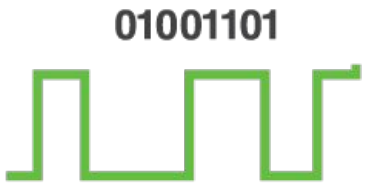


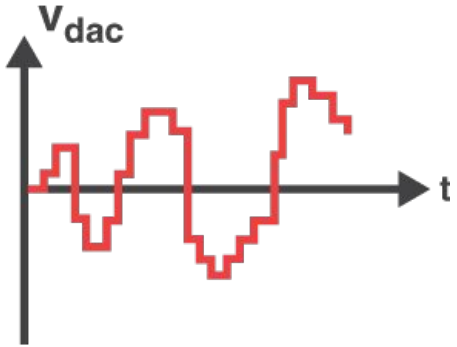


# Where does DSP fit?



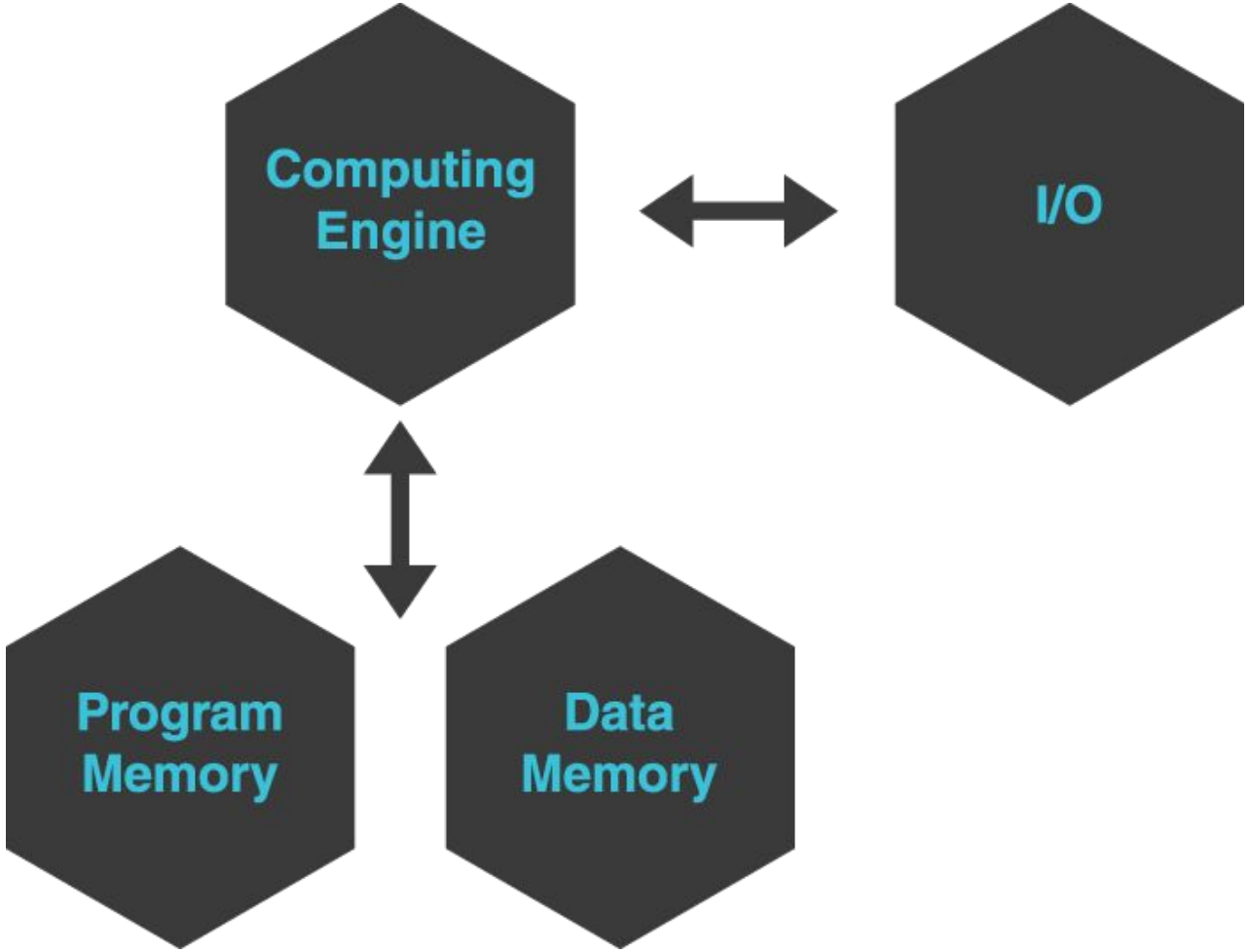



# DSPs have:

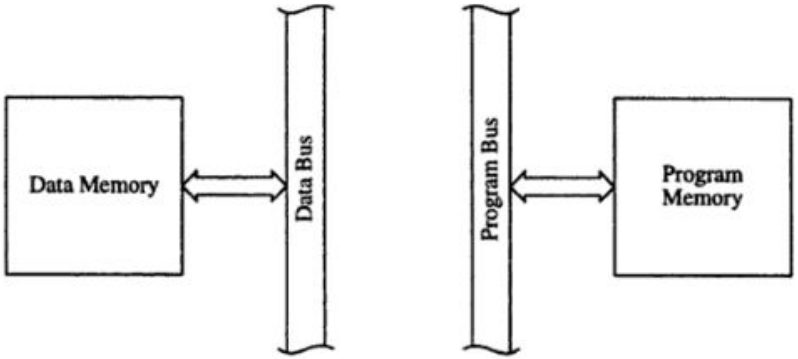
- Computing Engine
  - Mathematical manipulations and calculations
- Data Memory
  - stores the information to be processed
- Program Memory
  - stores the instructions that the DSP will use to process, compress, or manipulate data.
- I/O:
  - External ports, serial ports, timers- things that connect to the outside world.


Computing Engine
I/O
Program Memory
Data Memory

# Applications

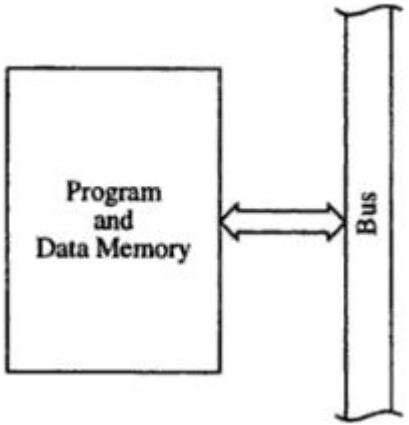
- Audio signal processing
- Speech recognition
- Digital image processing
- Speech coding and transmission in digital mobile phones
- Weather forecasting
- Economic forecasting
- Seismic data processing
- Medical imaging

# Harvard Architecture




- Can support pipelining- faster
- No complications and mix-ups between instruction and data
- Data and Instruction have a max capacity.

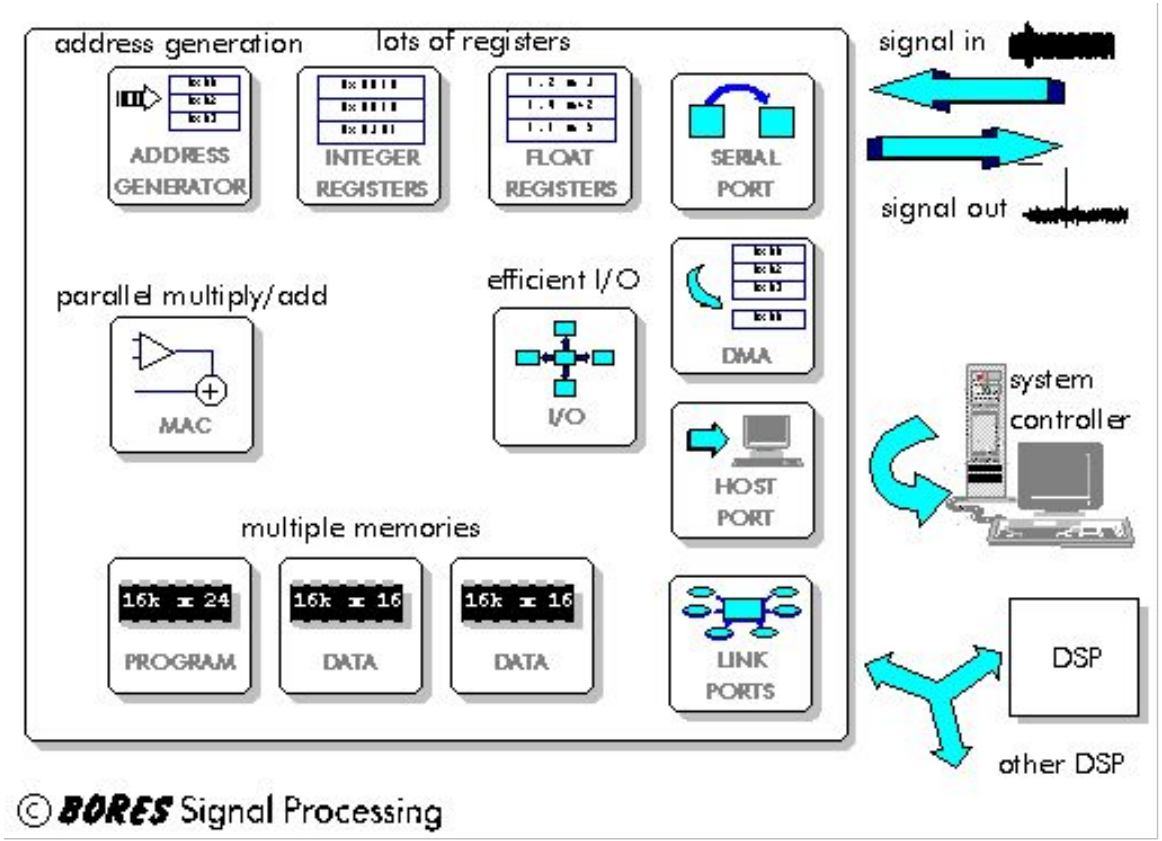
# Von Neumann Architecture



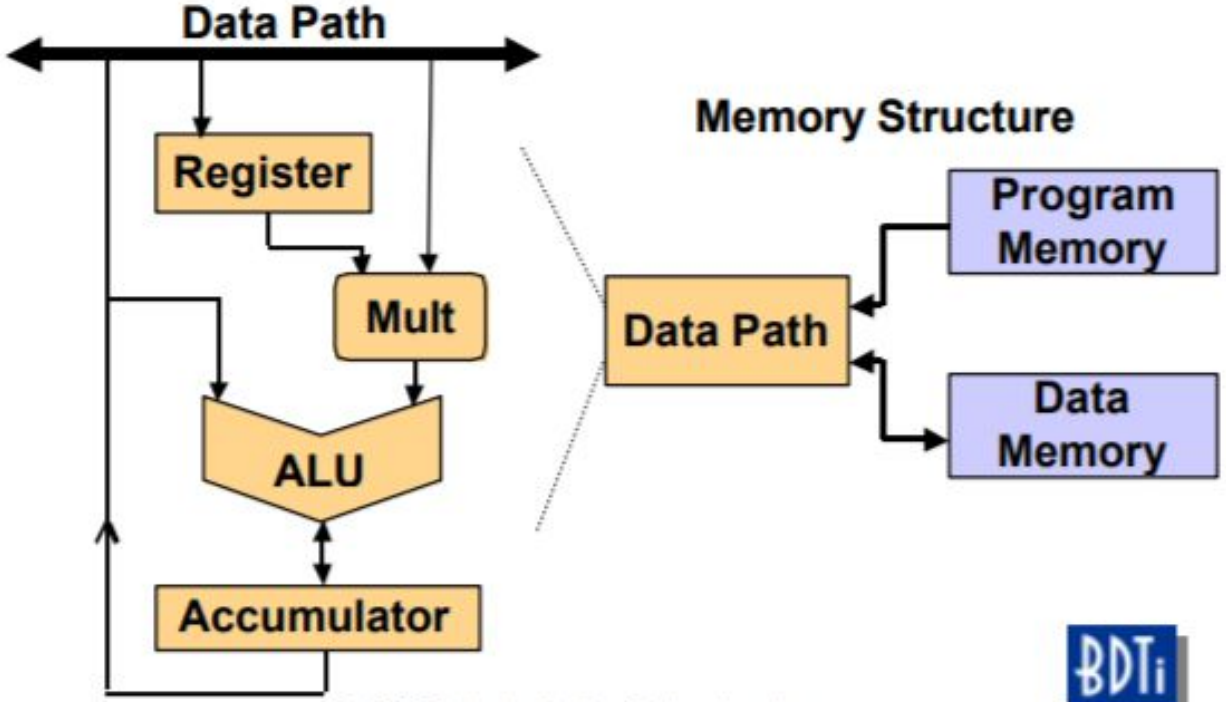

- Follows a linear fetch-decode-execute cycle
- Separate buses are not required
- Data space can be used for instructions, therefore no maximum ratio of division.

# What is a typical DSP made of?

# Differences from DSP and other GPUs



# Early DSP Architecture

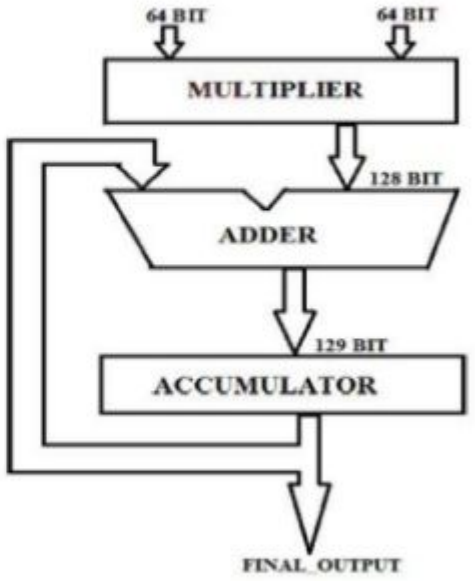


# Multiplier-Accumulator (MAC)

Multiplier obtains inputs from memory

Accumulator contains sum of previous products

Output of multiplier added to accumulator every clock cycle




Requires large amount of logic, so why?

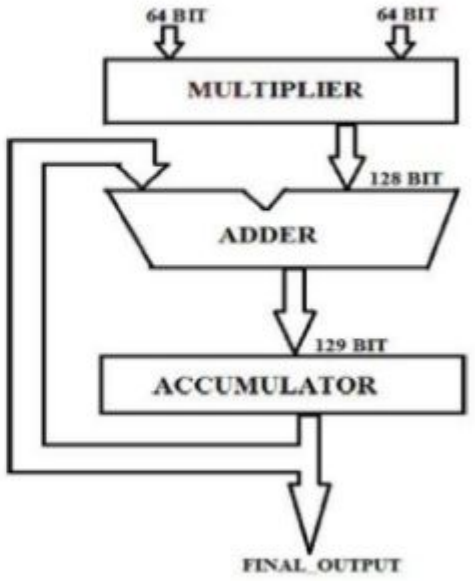
(Hint: every clock cycle)

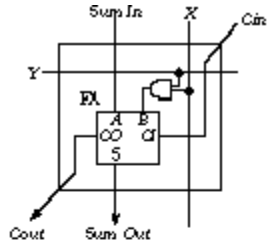
# Multiplier-Accumulator (MAC)

Compare to shift and add multiplication

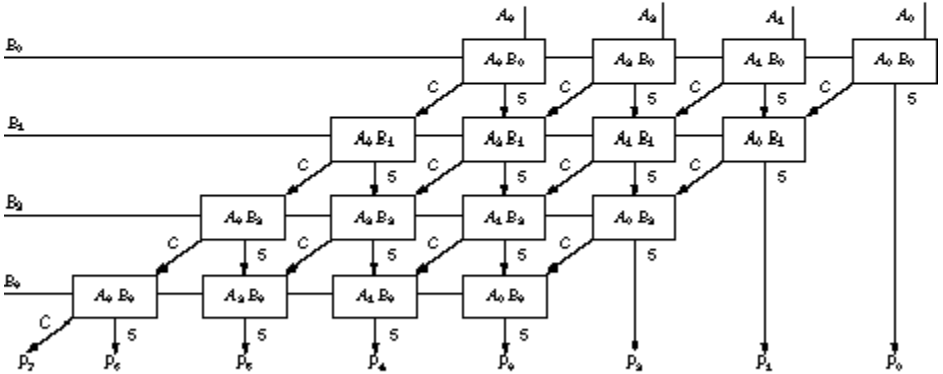
Speeds up and improves accuracy of computations that involve accumulating products

- Dot product
- Matrix multiplication
- Polynomial evaluation
- Convolutions
- Artificial neural networks





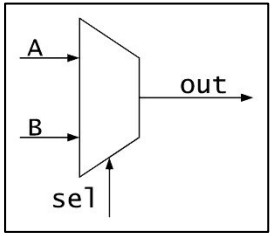

(a) Basic building block




(b) 4 × 4 multiplier structure

Figure 5.29 4 × 4 combinational multiplier .

# Multiplier-Accumulator (MAC)




Multiplier Accumulator

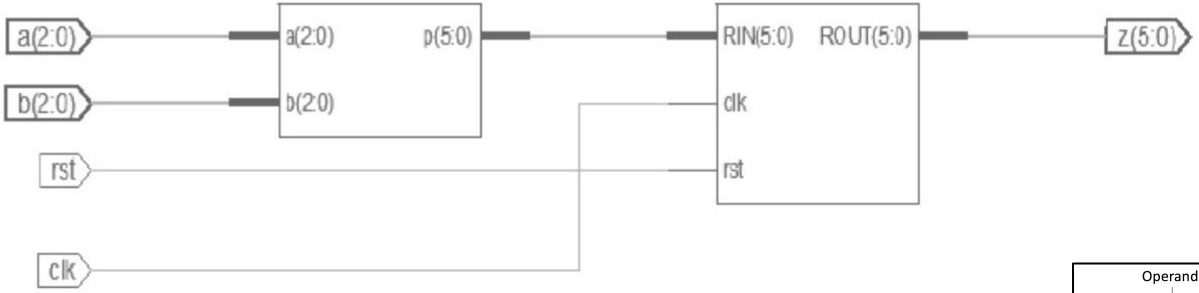
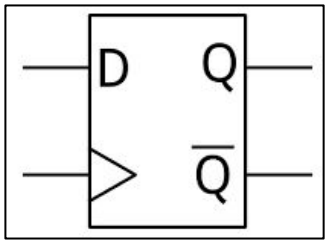
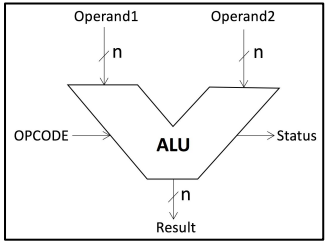



FIG 4.2 SCHEMATIC VIEW OF MAC UNIT



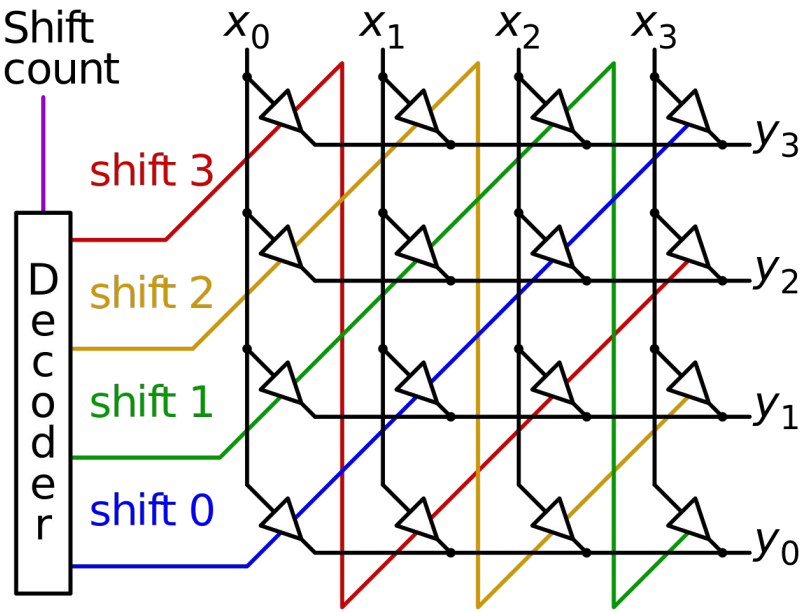
# Barrel Shifter

Pure combinational logic

Shift order of bits in 1 clock cycle

*n*-bit word ➔ $n \log_2 n$ = # muxes needed

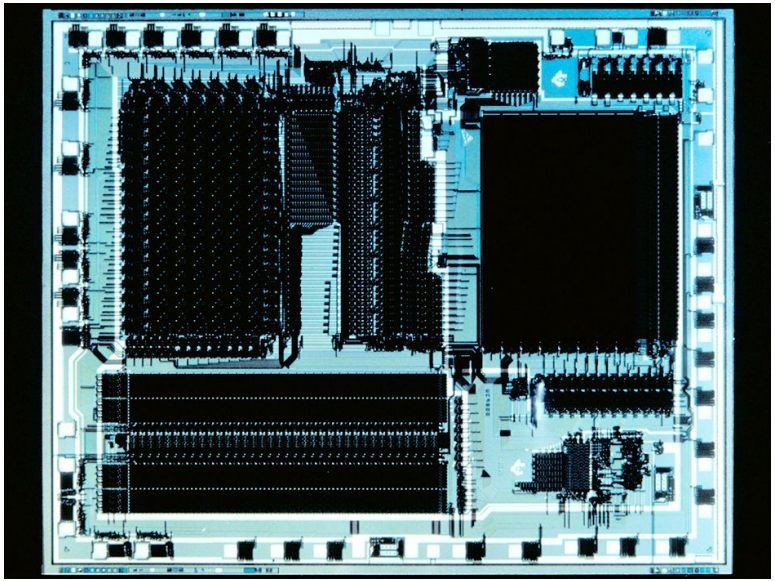
e.g., 16-bit ➔ $16 \log_2 16 = 16 * 4 = 64$ muxes

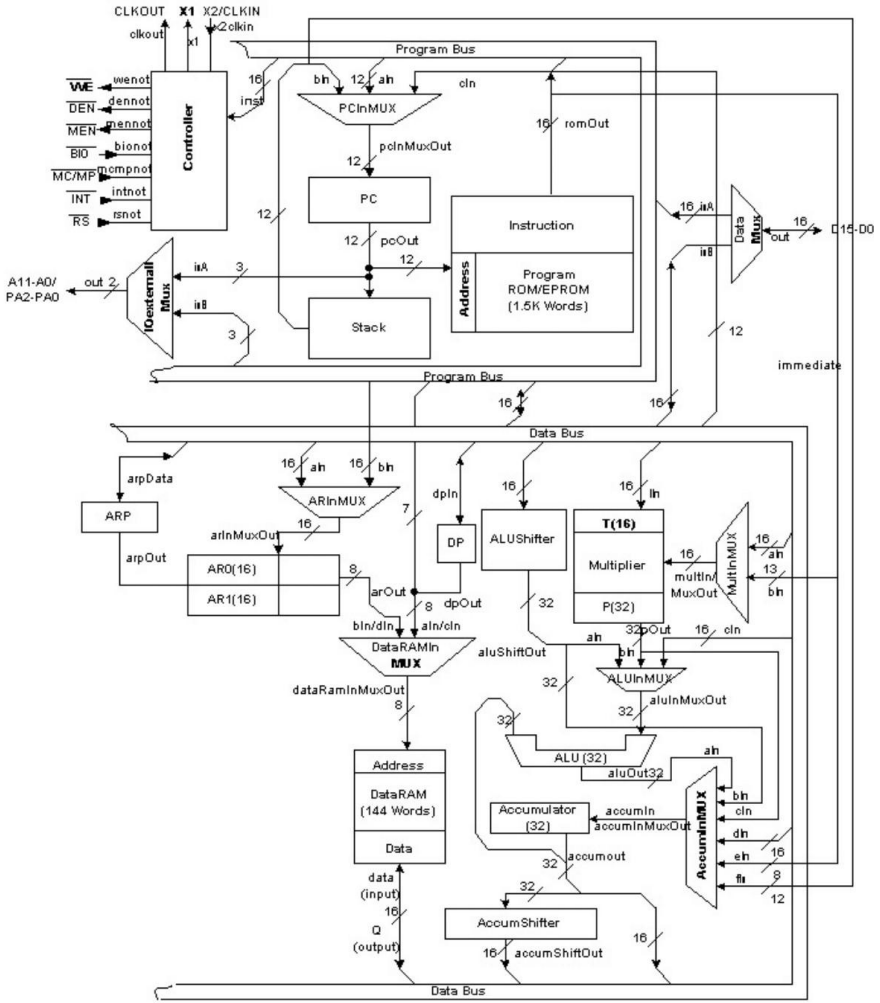


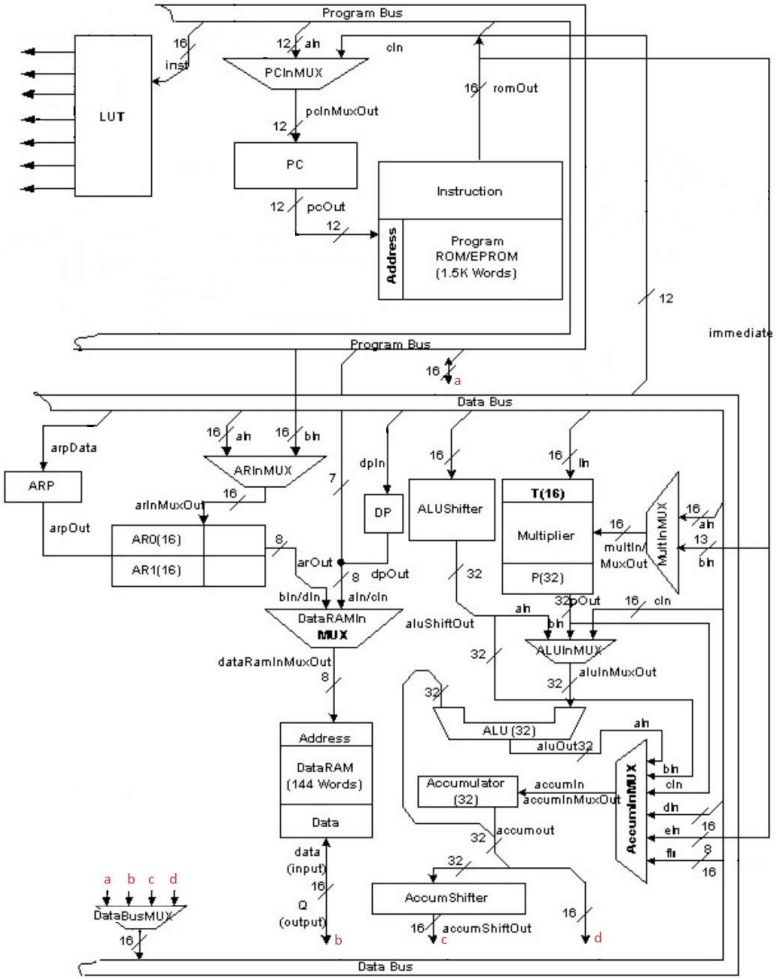
# TMS32010

# History

- Not the first DSP chip: April 8, 1983
- Fastest for its time: multiply operation in 200 nanosec
- It was also able to execute instructions from both on-chip ROM(Read Only Memory) and off-chip RAM (Random Access Memory).
- (The other chips at the time had only canned DSP functions)




CLKOUT
X1
X2/CLKIN
clkout
x1
x2clkin
Program Bus
WE
wenot
DEN
dennot
MEN
mennot
BIO
bionot
MC/MP
mcmpnot
INT
intnot
RS
rsnot
Controller
inst
16
bin
12
ain
cin
PCInMUX
pcInMuxOut
PC
pcOut
romOut
Instruction
Program
ROM/EPROM
(1.5K Words)
Address
Stack
iA
iB
iDestemalMux
A11-A0/
PA2-PA0
out
DataMux
D15-D0
immediate
Data Bus
ARP
arpData
arpOut
ARInMUX
arInMuxOut
AR0(16)
AR1(16)
arOut
dpOut
DP
dpIn
ALUShifter
T(16)
Multiplier
P(32)
multInMuxOut
MultInMUX
aluShiftOut
ALUInMUX
aluInMuxOut
ALU (32)
aluOut32
DataRAMIn
MUX
dataRamInMuxOut
bIn/dIn
aIn/cIn
Address
DataRAM
(144 Words)
Data
data
(input)
Q
(output)
Accumulator
(32)
accumIn
accumInMuxOut
accumout
AccumInMUX
AccumShifter
accumShiftOut

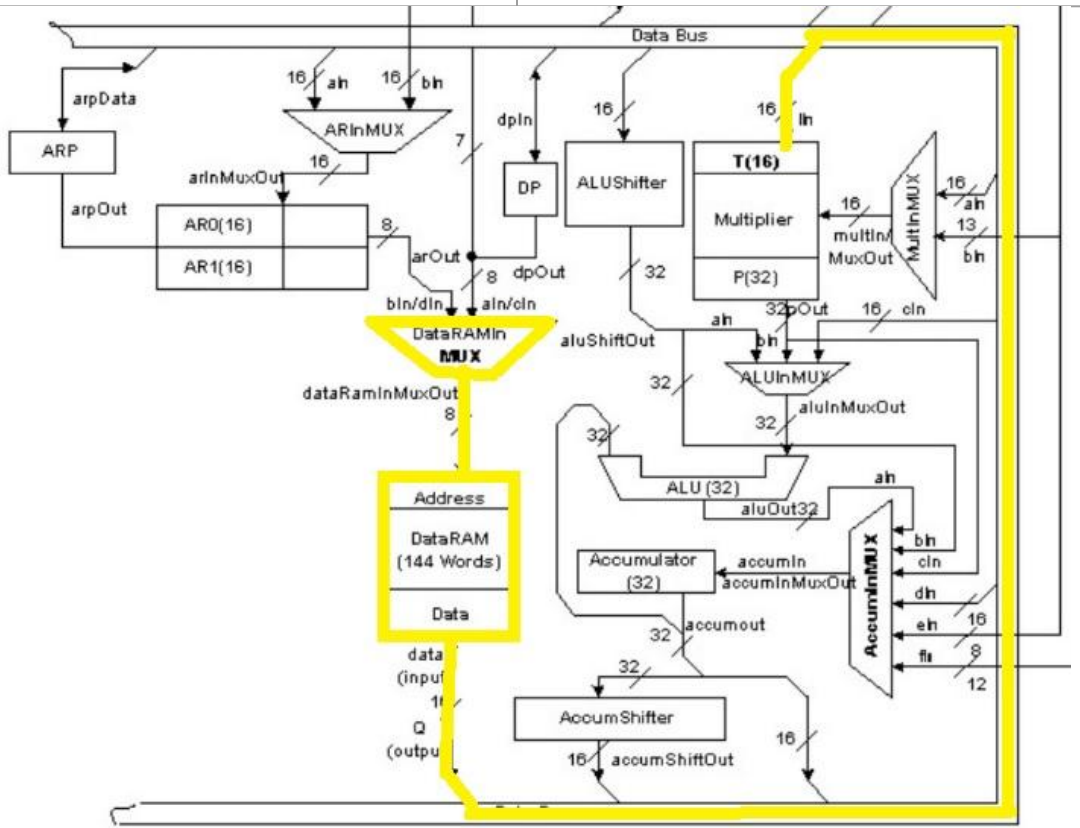

Program Bus
16
inst
LUT
12
ah
cln
PCInMUX
pcInMuxOut
12
PC
12
pcOut
12
16
romOut
Instruction
Address
Program
ROM/EPROM
(1.5K Words)
12
immediate
Program Bus
16
a
Data Bus
arpData
ARP
arpOut
16
ah
16
bh
ARInMUX
arInMuxOut
16
AR0(16)
AR1(16)
8
arOut
8
dpOut
7
dpIn
DP
16
ALUShifter
32
16
lh
T(16)
Multiplier
P(32)
16
multIn/
MuxOut
MultInMUX
16
ah
13
bh
bIn/dIn
aIn/cIn
DataRAMIn
MUX
aluShiftOut
32pOut
16
cln
ah
bh
ALUInMUX
dataRamInMuxOut
8
32
aluInMuxOut
32
32
ALU (32)
aluOut32
ah
Address
DataRAM
(144 Words)
Data
Accumulator
(32)
accumIn
accumInMuxOut
bh
cln
dh
eh
fh
16
8
16
AccumInMUX
accumout
32
data
(input)
16
Q
(output)
b
32
AccumShifter
16
accumShiftOut
c
16
d
a
b
c
d
DataBusMUX
16
Data Bus

# Small Group Work

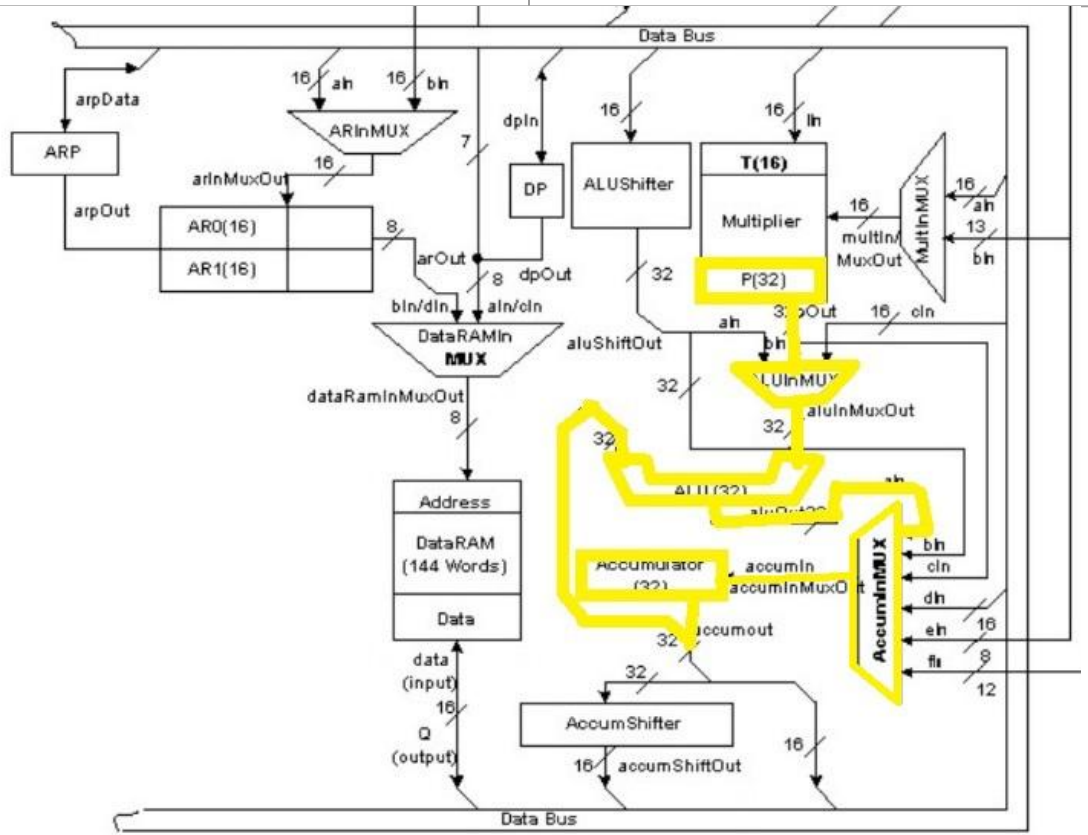
# Small group work: Data Path

| **Instruction** | **Operation** |
| --- | --- |
| LT: Load T-Register | (dma) → T |
| APAC : Add P-Register to Accumulator | (ACC)+(P)→ACC |

| Instruction | Operation |
| --- | --- |
| LT: Load T-Register | (dma) → T |



| Instruction | Operation |
| --- | --- |
| APAC : Add P-Register to Accumulator | (ACC)+(P)→ACC |



# Questions?

## Resources Utilized:


https://archive.org/details/bitsavers_tiTMS320xx985_13292501
http://meseec.ce.rit.edu/eecc722-fall2003/722-10-8-2003.pdf
https://www.slideshare.net/Abhishekt11/dsp-datapath
https://en.wikipedia.org/wiki/Barrel_shifter
https://www.intel.com/content/www/us/en/programmable/support/support-resources/design-examples/design-software/verilog/ver-unsigned-multiply-accumulator.html
https://www.scribd.com/doc/24177561/Verilog-Code-for-Mac-Unit
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.567.1157&rep=rep1&type=pdf
http://www.ti.com/lit/ds/symlink/tms320c25.pdf
https://en.wikipedia.org/wiki/Multiply%E2%80%93accumulate_operation
https://spectrum.ieee.org/tech-history/silicon-revolution/chip-hall-of-fame-texas-instruments-tms32010-digital-signal-processor
https://en.wikipedia.org/wiki/Digital_signal_processing
https://www.allaboutcircuits.com/technical-articles/an-introduction-to-digital-signal-processing/