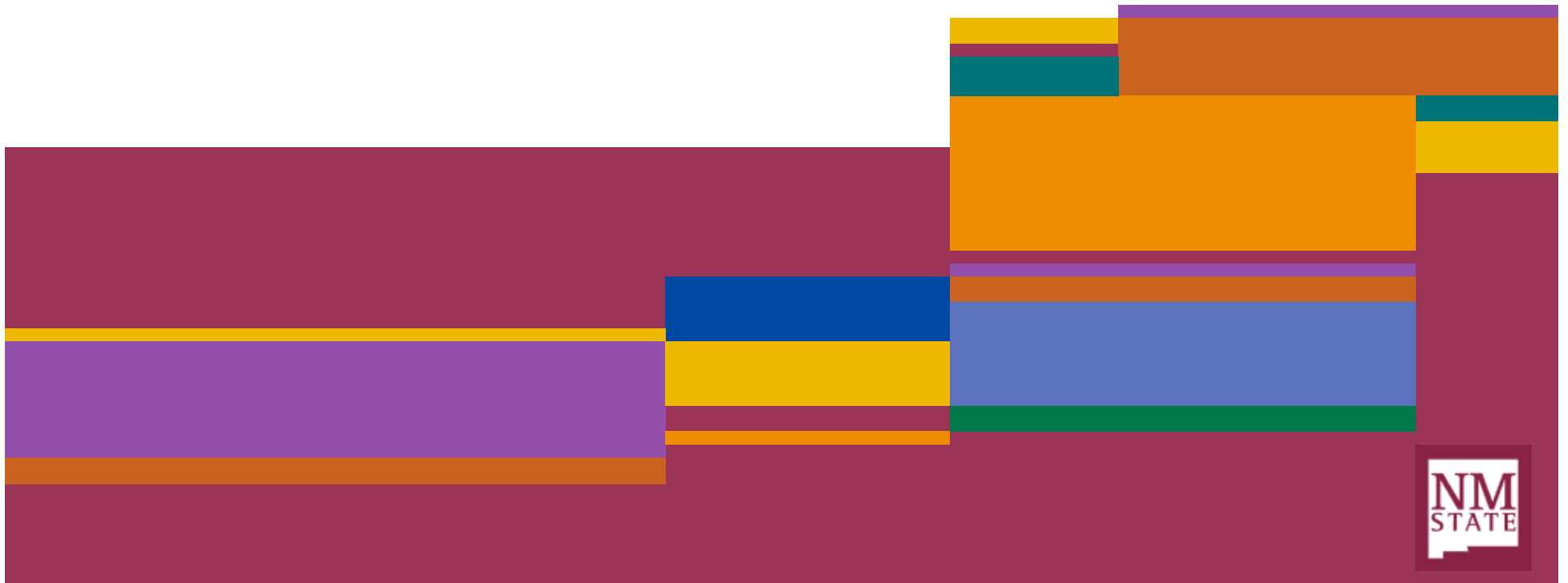
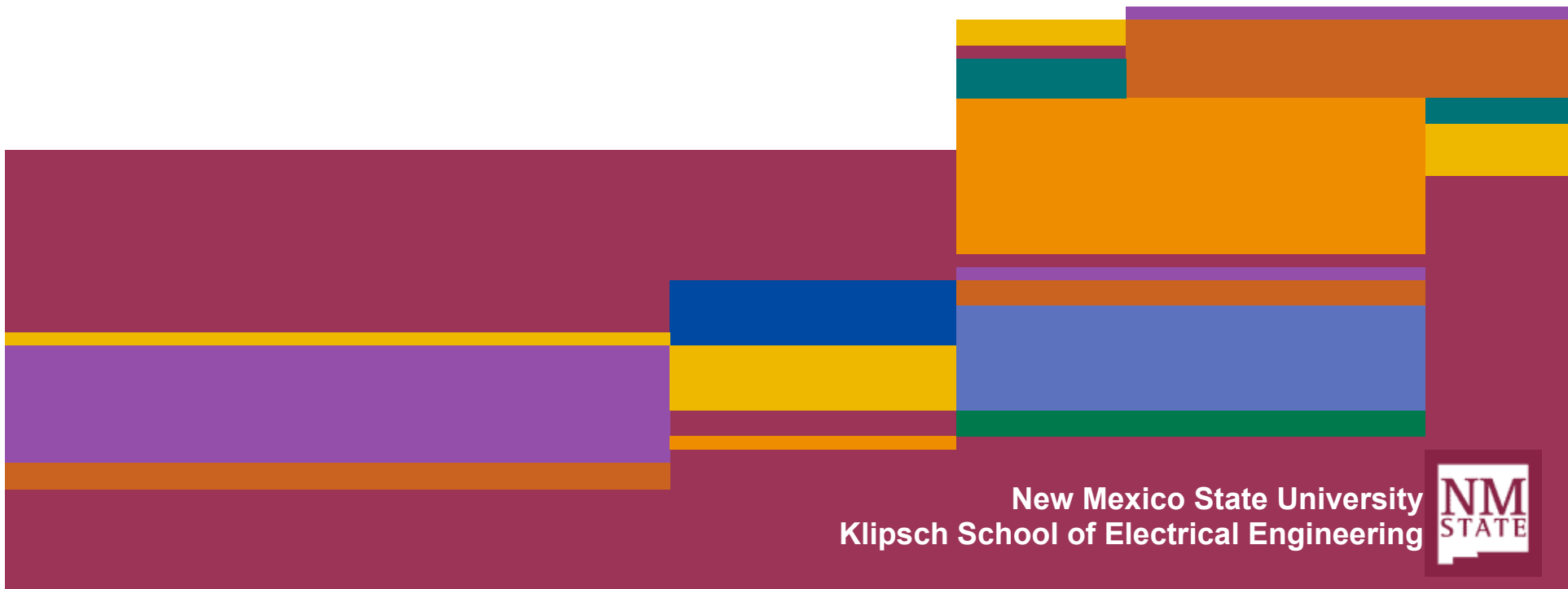


Introduction to the Texas Instruments TMS320C6416 and DSP Starter Kit (6416DSK)



Introduction to TI DSP



New Mexico State University
Klipsch School of Electrical Engineering

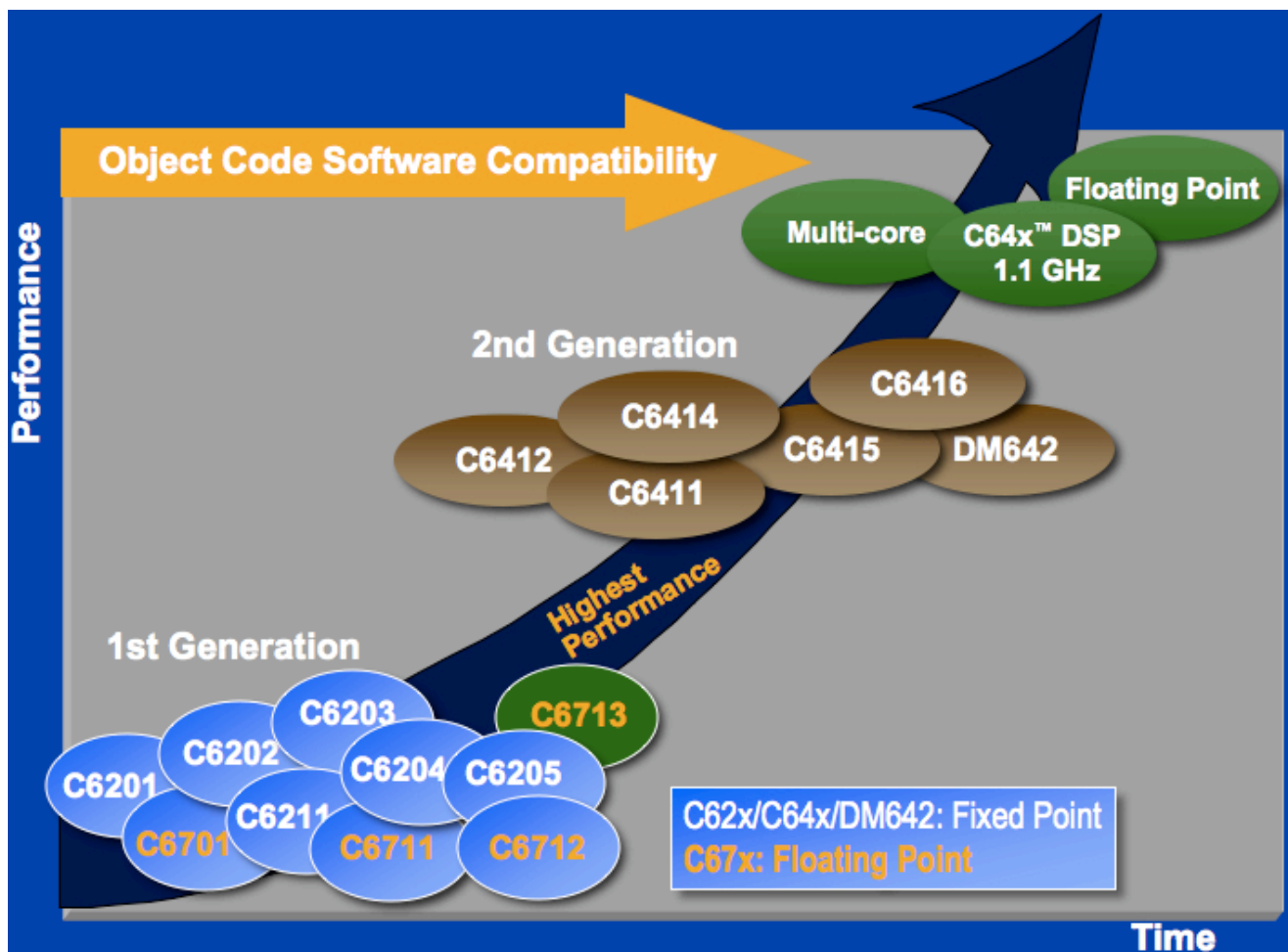


Texas Instruments TMS320 Family

- Various DSP families/sub-families for different markets



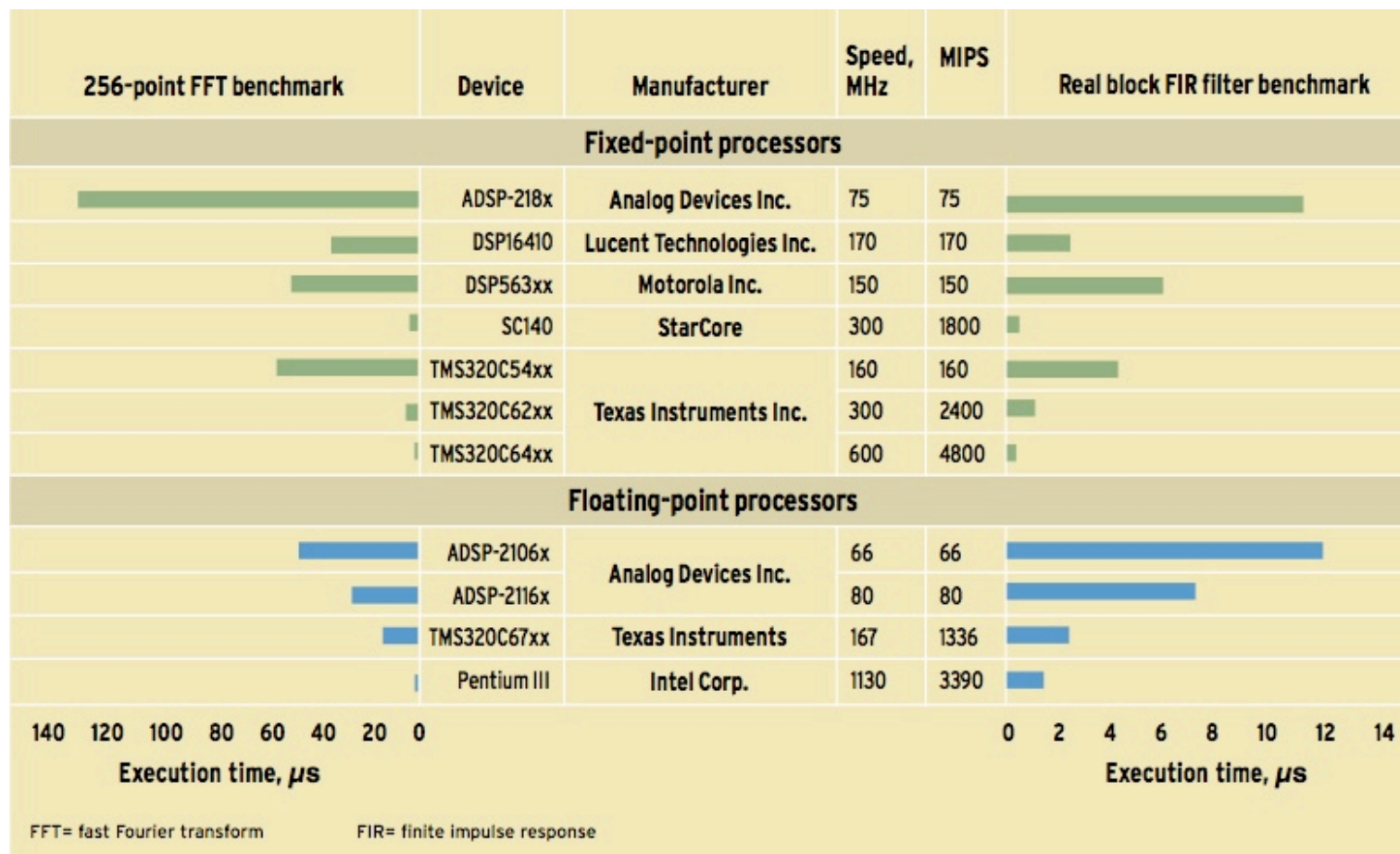
C6000 Roadmap



Fixed-Point vs. Floating-Point Processors

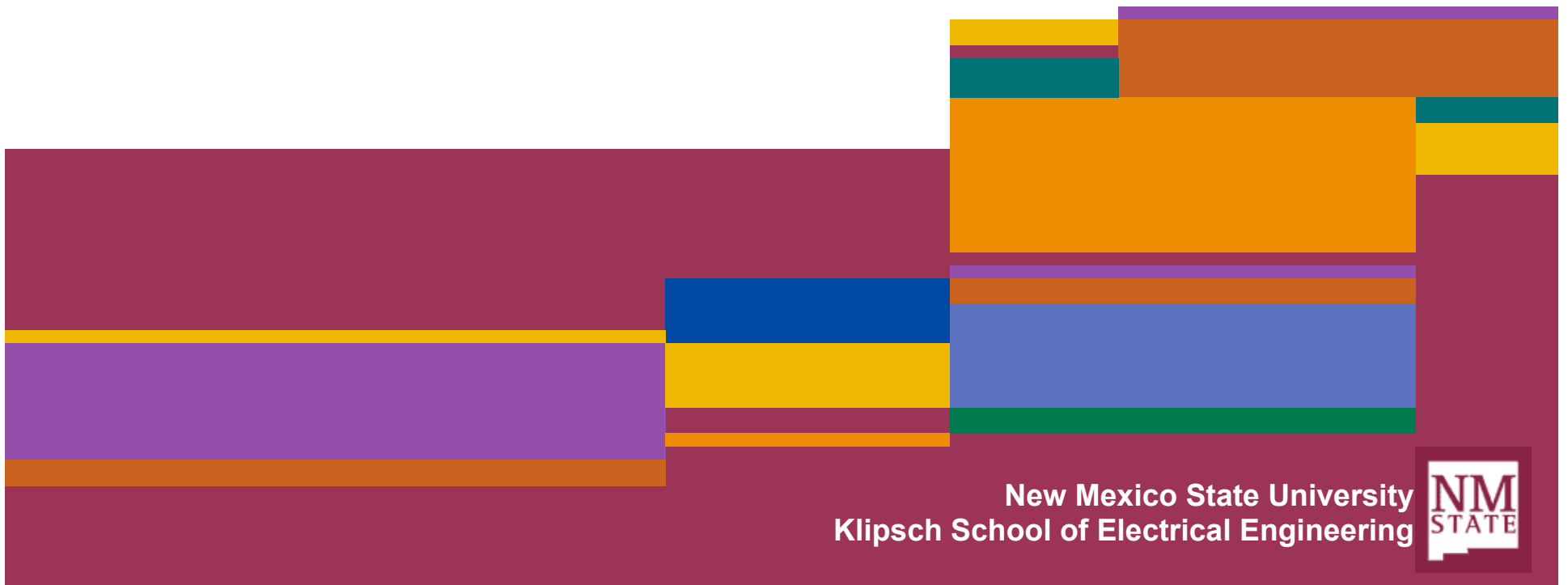
- Floating-point chosen for applications requiring:
 - High numerical precision, wide dynamic range
 - Easier to program
- Drawbacks of floating-point DSP
 - More expensive, higher power consumption, and larger package size compared to fixed-point DSP
- Application dictates which DSP type
- Learning to program for floating-point targets is trivial if one has programmed fixed-point targets

DSP Benchmarks (2001)



Shorter bars indicate faster processing

VLIW Architecture



Improving DSP Performance

- DSPs pack *many* operations into a *single* instruction

```
macr -x0,y0,a    x:(r0)+,x0    y:(r4)+,y0
```

- Processor performance can be improved with a *multi-issue* approach
 - Use *simple instructions* that encode a single operation
 - *Issue and execute multiple instructions* in parallel groups so as to achieve high level of parallelism
- Two classes of multi-issue architectures: Very Long Instruction Word (VLIW) and superscalar

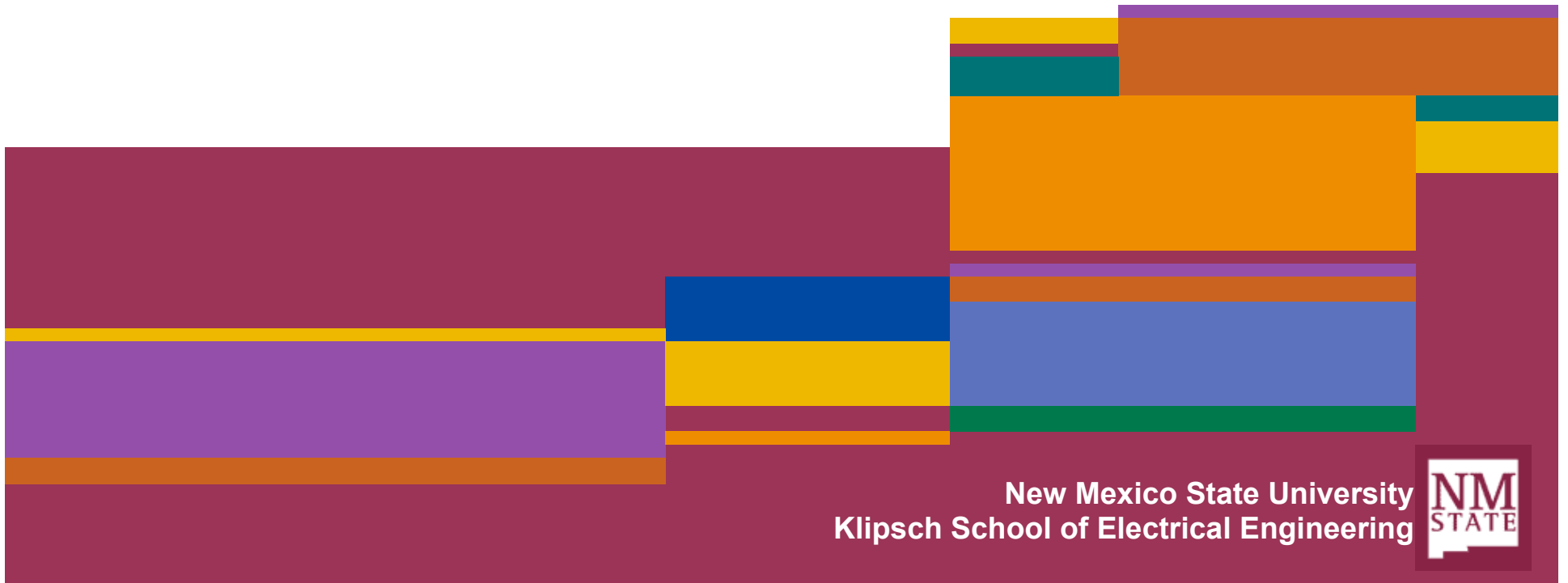
VLIW and Superscalar Architectures

- Very Long Instruction Word (VLIW) architecture
 - Provides several functional units each of which executes an instruction
 - *Programmer* (or compiler) *specifies* which instructions are executed in parallel (and on which functional units) and are grouped into VLIWs (256 bits)
 - TMS320C64xx can perform up to *eight* instructions in a single clock cycle
- Superscalar
 - *Processor determines* which instructions are executed in parallel based on data dependencies and resource contention (dynamic scheduling of parallel ops)
 - Not typically used in DSPs because of difficulty in predicting execution time (important in real-time applications)

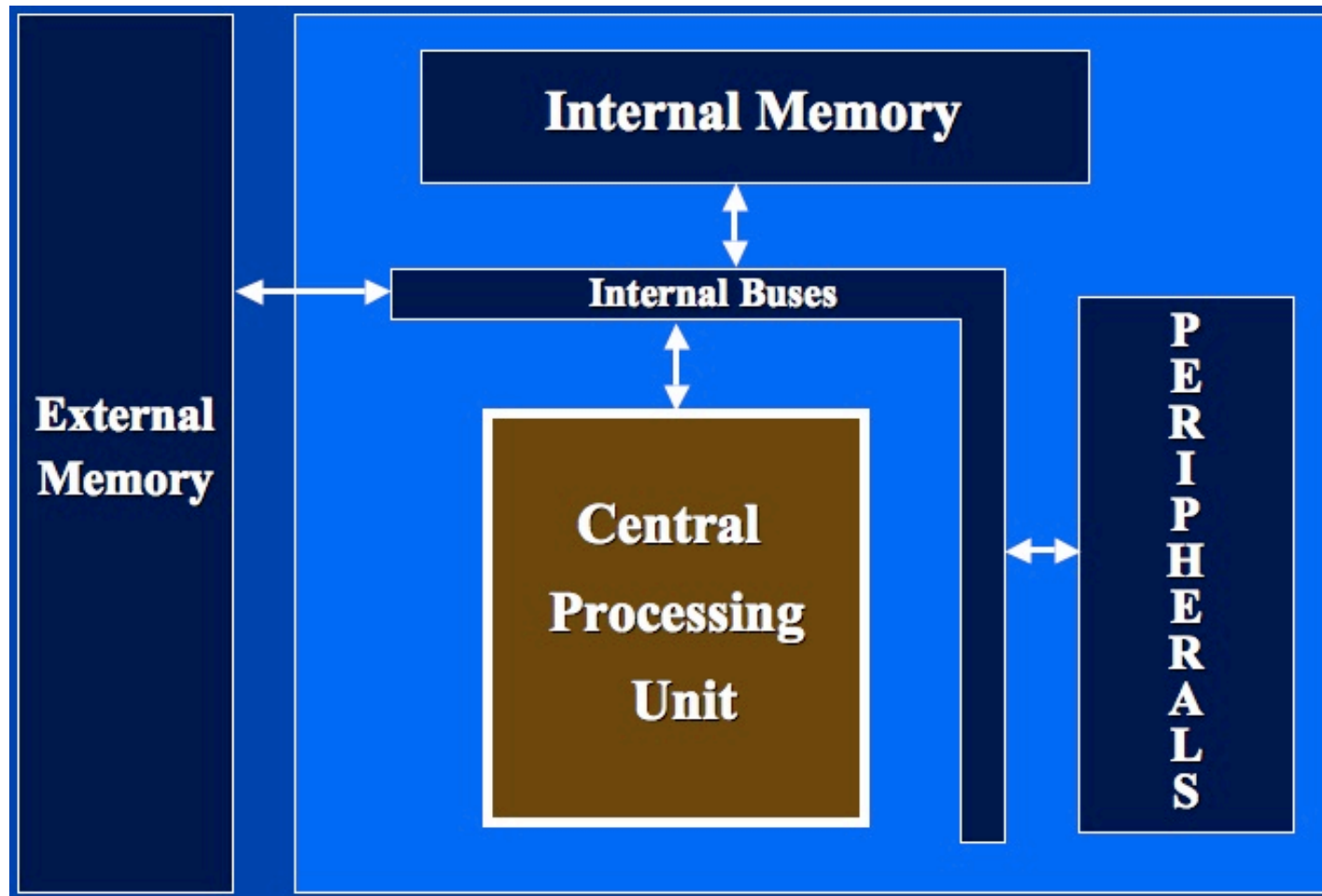
VLIW Architecture

- **Architectural complexity issues**
 - Increased cost and power consumption of VLIW largely offset by increased performance
 - Very tricky assembly programming -- must keep track of multiple execution units and schedule multiple instructions for parallel execution
 - Advanced compiler designs minimize programming difficulty (since instructions are simple) but object code is generally not as efficient as hand-coded assembly
- **Need for large I/O (memory) bandwidth**
 - Need External Memory Interface (EMIF) which operates at high speed (C6416 EMIFA operates at 133MHz)
 - Wider data buses allow retrieval of more data words per clock cycle to keep multiple execution units fed with data

C64x Features



General DSP System Block Diagram



C64x Key Features

- CPU features

- Sixty-four 32-bit general purpose (GP) registers arranged in register files “A” and “B”
- CPU is able to perform four 16-bit multiplications in parallel
- LOAD on C6416 allows address pointer to be pre- or post-modified as part of instruction
- 11-stage non-interlocked pipeline (2 - 5 cycle latency). Multiplications, for example, have two-cycle latencies; Data loads have five-cycle latencies. Both have single-cycle throughput, however

C64x Key Features (cont' d)

- On-chip memory implements a Harvard architecture-- separate program and data memories
 - Program memory -- 32-bit address bus and 256-bit data bus
 - Data memory -- two 32-bit address buses and two 64-bit data buses
 - Uses a two-level on-chip memory organization, with L1 program and data caches and a unified (program and data) L2 cache
- Supports modulo addressing and can maintain eight concurrently active circular buffers

C64x Key Features (cont' d)

- **Peripherals**

- All C64x family members include a 64-channel enhanced DMA controller, two or three multi-channel buffered serial ports (McBSP), and three 32-bit timers
- C64x chips may also include host ports, PCI interfaces, serial audio ports, I2C ports, video ports, Ethernet MACs, Viterbi decoder coprocessors, turbo decoder coprocessors, and UTOPIA ports

- **Power consumption (varies by family member)**

- TMS320C6414T consumes 673 mW at 600 MHz and 1.1 volts (based on a typical DSP workload)

- **Cost (2009)**

- TMS320C6416 at 600 MHz ~\$100 (1000+)
- TMS320C6416T at 1 GHz ~\$250 (1000+)

Source: www.bdti.com

C64x Core, Instruction Set, and Cross Paths

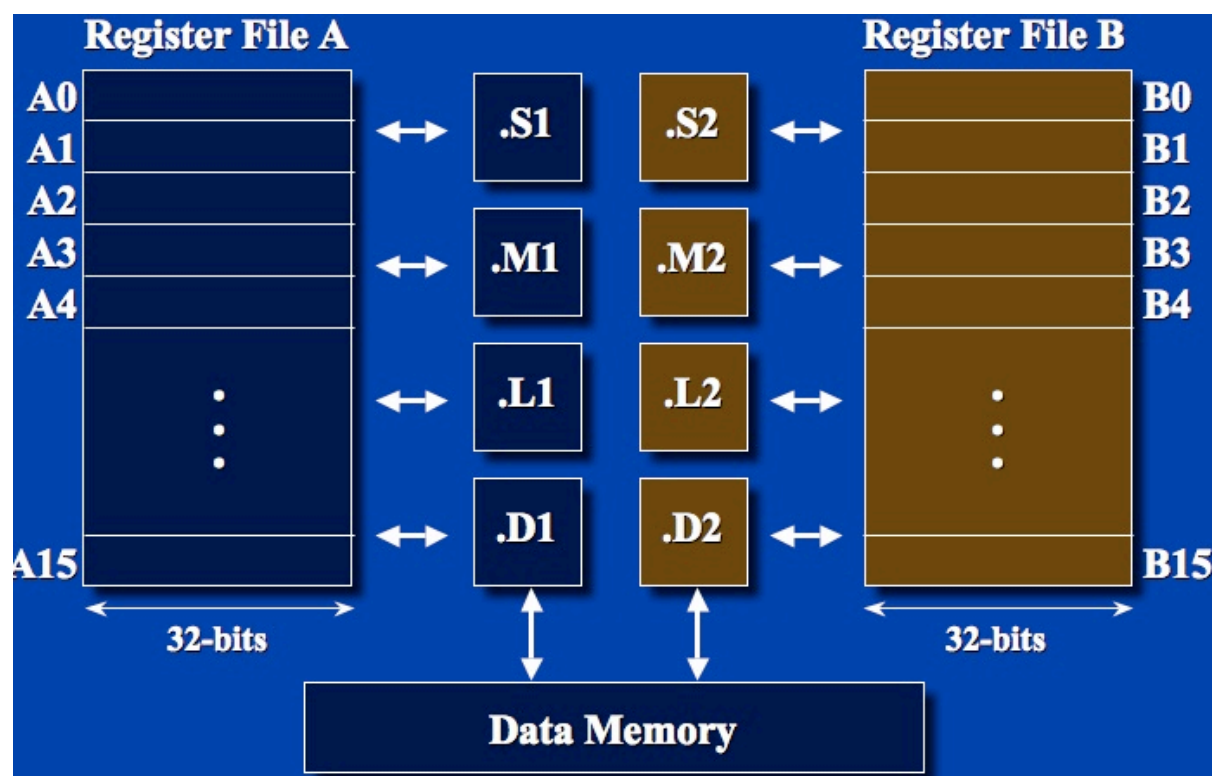


C6000 Core Functional Units

- **.M unit (multiplier unit performs multiplications)**
 - Two 16x16 multiply operations
 - 16x32 multiply operations
- **.L unit (logic unit)**
 - 32-bit logical operations
 - 32/40-bit arithmetic and compare operations
- **.S unit (shift unit and branching)**
 - 32/40-bit shifts; branches
- **.D unit (data loading unit)**
 - Loading/storing from data cache
 - 32-bit add, subtract, linear and circular address calculation
 - LDB: loads one byte (8-bit), LDH: loads half word (16-bit), LDW: loads a word (32-bit), LDDW: loads a double word (64-bit)

C6000 Core Functional Units (cont' d)

- To increase throughput, DSP has “A and B sides” also labeled as “1 and 2”



- 8 functional units: S1, S2, M1, M2, L1, L2, D1, and D2
- 32 registers: File A (A0 - A15) and File B (B0 - B15)

C62x Instruction Set

.S Unit	
ADD	MVKLH
ADDK	NEG
ADD2	NOT
AND	OR
B	SET
CLR	SHL
EXT	SHR
MV	SSHL
MVC	SUB
MVK	SUB2
MVKL	XOR
MVKH	ZERO

.L Unit	
ABS	NOT
ADD	OR
AND	SADD
CMPEQ	SAT
CMPGT	SSUB
CMPLT	SUB
LMBD	SUBC
MV	XOR
NEG	ZERO
NORM	

.D Unit	
ADD	STB/H/W
ADDA	SUB
LDB/H/W	SUBA
MV	ZERO
NEG	

.M Unit	
MPY	SMPY
MPYH	SMPYH

Other	
NOP	IDLE

- C64x instruction set (2nd generation) is based on C62x instruction set (1st generation)
- Each functional unit can execute certain instructions
- Some instructions can be executed on multiple functional units, i.e. multiple ADDs in parallel

C64x Instruction Set (Superset of C62x)

.S	<u>Dual/Quad Arith</u> SADD2 SADDUS2 SADD4	<u>Data Pack/Un</u> PACK2 PACKH2 PACKLH2 PACKHL2 UNPKHU4 UNPKLU4 SWAP2 SPACK2 SPACKU4	<u>Compares</u> CMPEQ2 CMPEQ4 CMPGT2 CMPGT4	.L	<u>Dual/Quad Arith</u> ABS2 ADD2 ADD4 MAX MIN SUB2 SUB4 SUBABS4	<u>Data Pack/Un</u> PACK2 PACKH2 PACKLH2 PACKHL2 PACKH4 PACKL4 UNPKHU4 UNPKLU4 SWAP2/4
	<u>Bitwise Logical</u> ANDN	<u>Branches/PC</u> BDEC BPOS BNOP ADDKPC	<u>Bitwise Logical</u> ANDN		<u>Multiplies</u> MPYHI MPYLI MPYHIR MPYLIR MPY2 SMPY2	
.D	<u>Dual Arithmetic</u> ADD2 SUB2	<u>Mem Access</u> LDDW LDNW LDNDW STDW STNW STNDW	.M	<u>Average</u> AVG2 AVG4	<u>Bit Operations</u> BITC4 BITR DEAL SHFL	DOTP2 DOTPN2 DOTPRSU2 DOTPNRSU2 DOTPU4 DOTPSU4 GMPY4 XPND2/4
	<u>Bitwise Logical</u> AND ANDN OR XOR	<u>Load Constant</u> MVK (5-bit)		<u>Shifts</u> ROTL SSHVL SSHVR	<u>Move</u> MVD	
	<u>Address Calc.</u> ADDAD					

C6000 Assembly for FIR Filtering

$$y[n] = \sum_{k=0}^{39} h_k x[n-k]$$

```

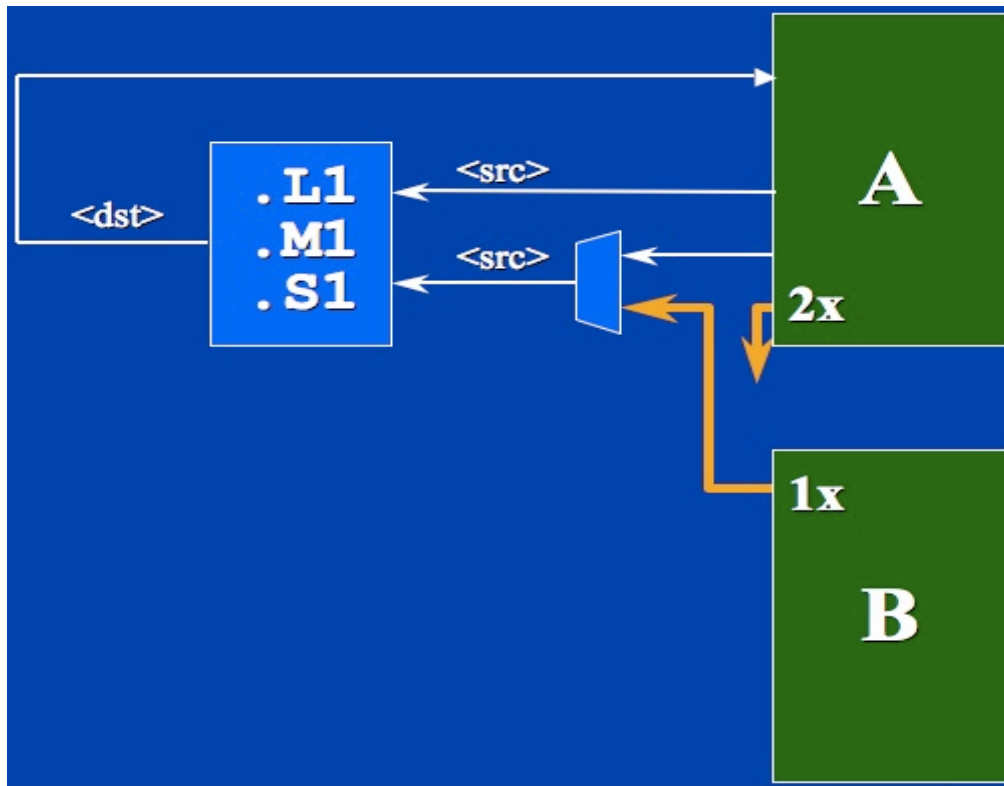
loop:   MVK    .S1    40,A2           ;A2 = 40, loop count
        LDH    .D1    *A5++,A0       ;A0 = h(k), 16 bit coeffs
        LDH    .D1    *A6++,A1       ;A1 = x(n-k), 16 bit samples
        MPY    .M1    A0,A1,A3       ;A3 = h(k) * x(n-k)
        ADD    .L1    A3,A4,A4       ;y[n] = y[n] + A3, accumulate
        SUB    .L1    A2,1,A2        ;decrement loop count
[A2]    B       .S1    loop           ;if A2 <> 0, branch
        STH    .D1    A4,*A7         ;*A7 = y[n]
```

– Instruction – Functional Unit – Operands

Cross Paths

- To exchange operands between the “A” and “B” sides, *cross paths* are used to link the sides
- Two types of cross paths are “data” and “address”

Data Cross Paths



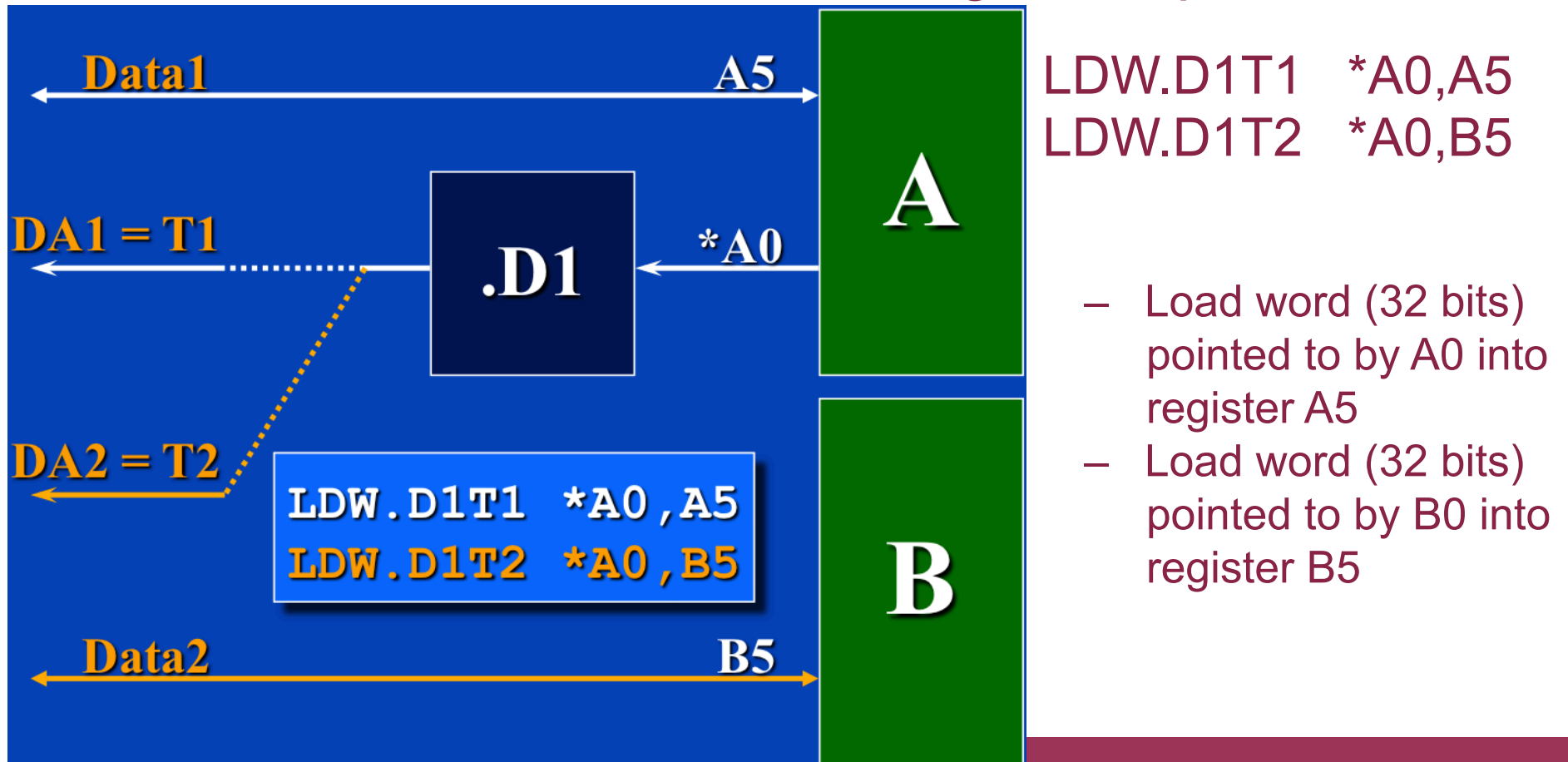
- Data cross paths allow operands from one side to be used by the other side
 - “1X” conveys data from side B to side A and “2X” conveys from A to B

```
ADD .L1x A0,A1,B2
MPY .M1x A0,B6,A9
```

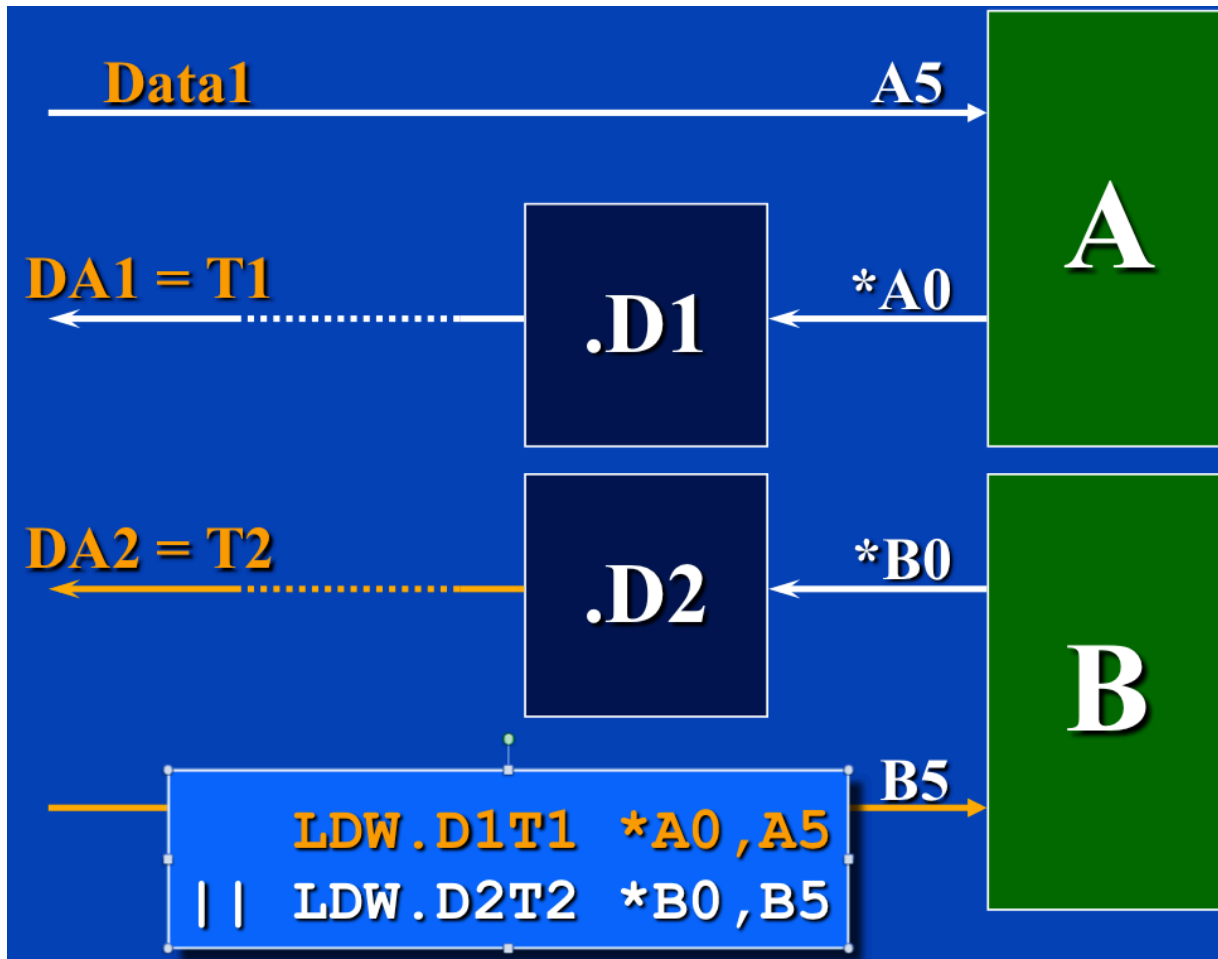
– Instruction – Functional Unit – Operands

Address Cross Paths

- Load or store to either side using cross paths



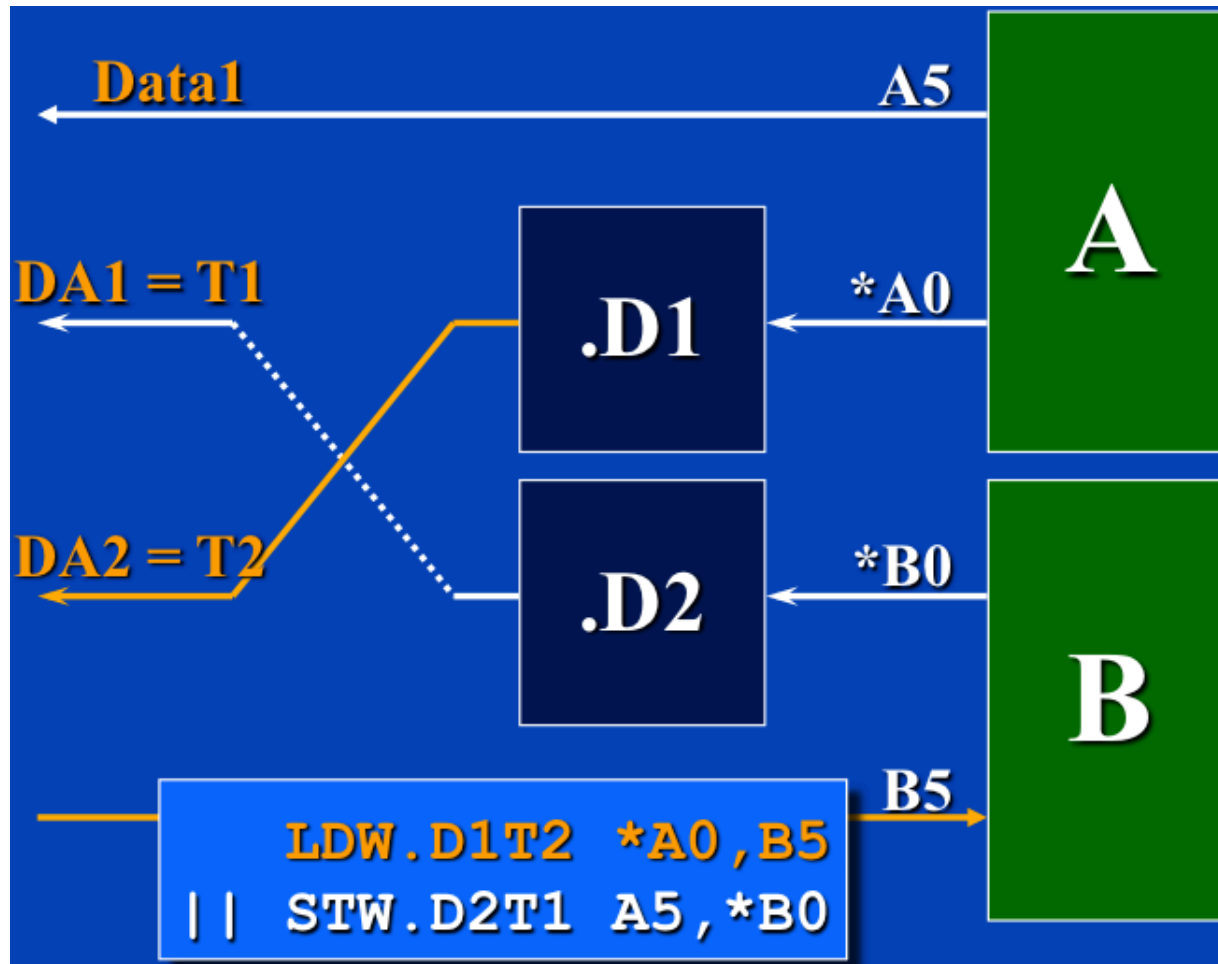
Standard Parallel Loads



```
LDW.D1T1 *A0,A5
|| LDW.D2T2 *B0,B5
```

- Load word (32 bits) pointed to by A0 into register A5
- Load word (32 bits) pointed to by B0 into register B5

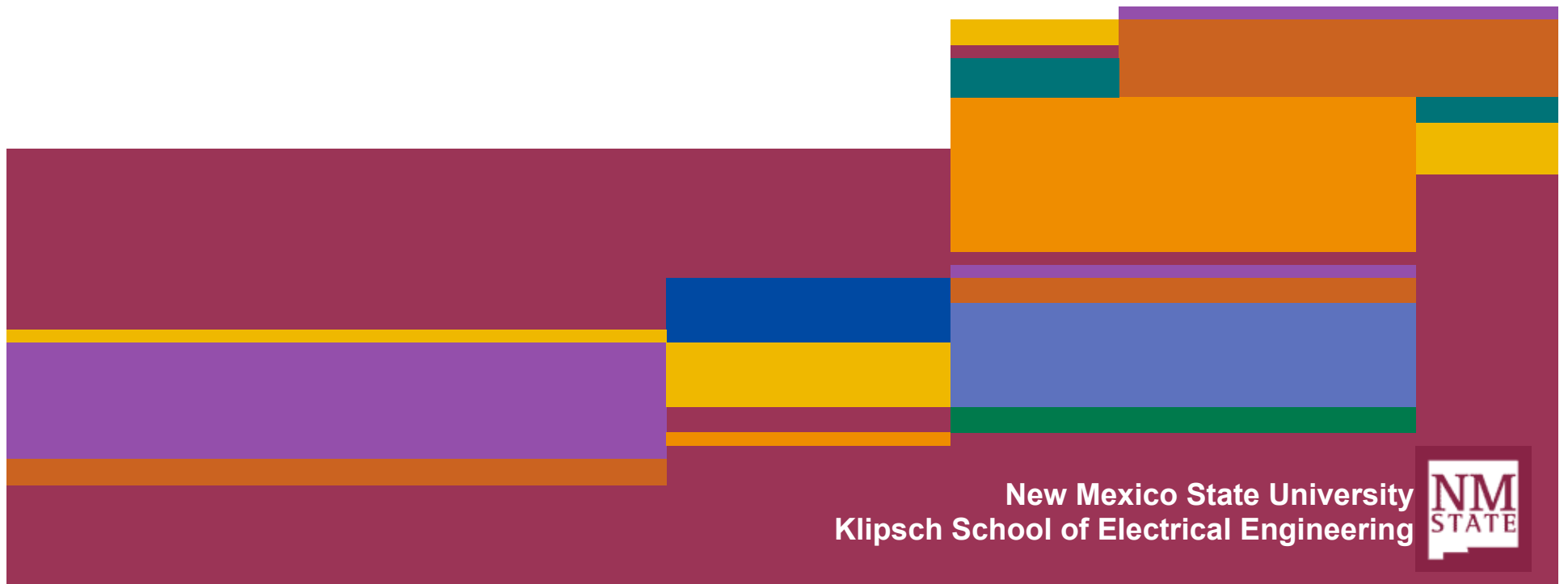
Parallel Loads using Address Cross Paths



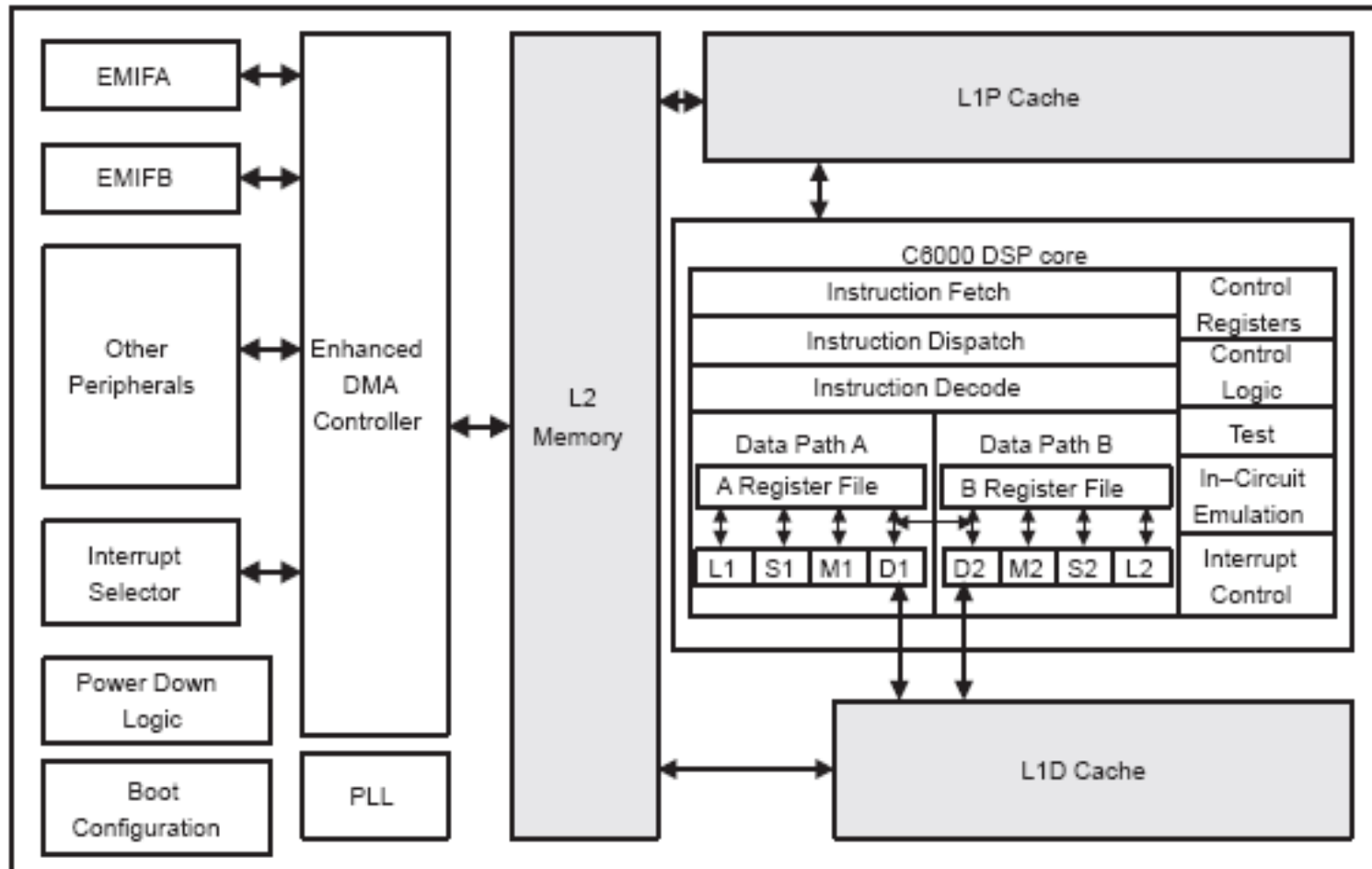
LDW.D1T2 *A0, B5
 || STW.D2T1 A5, *B0

- Load word (32 bits) pointed to by A0 into register B5
- Store word (32 bits) in register A5 to location pointed to by B0

C64xx Internal and External Memory



TMS320C6416 Block Diagram

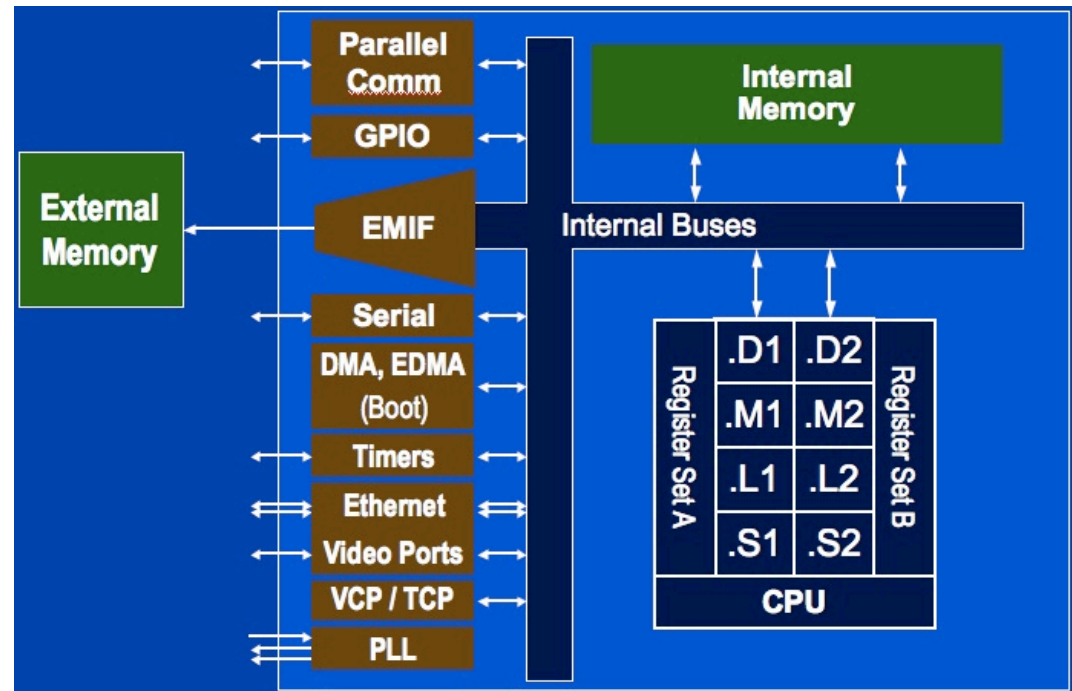


Internal Memory

- Two–Level Cache Architecture
- Level 1 Cache:
 - L1P: level–one program cache (16KB) (Direct mapped)
 - L1D: level–one data cache (16KB) (2-way set associative)
- Level 2 Cache:
 - L2: Unified program/data cache (1024KB) (4-way set associative)

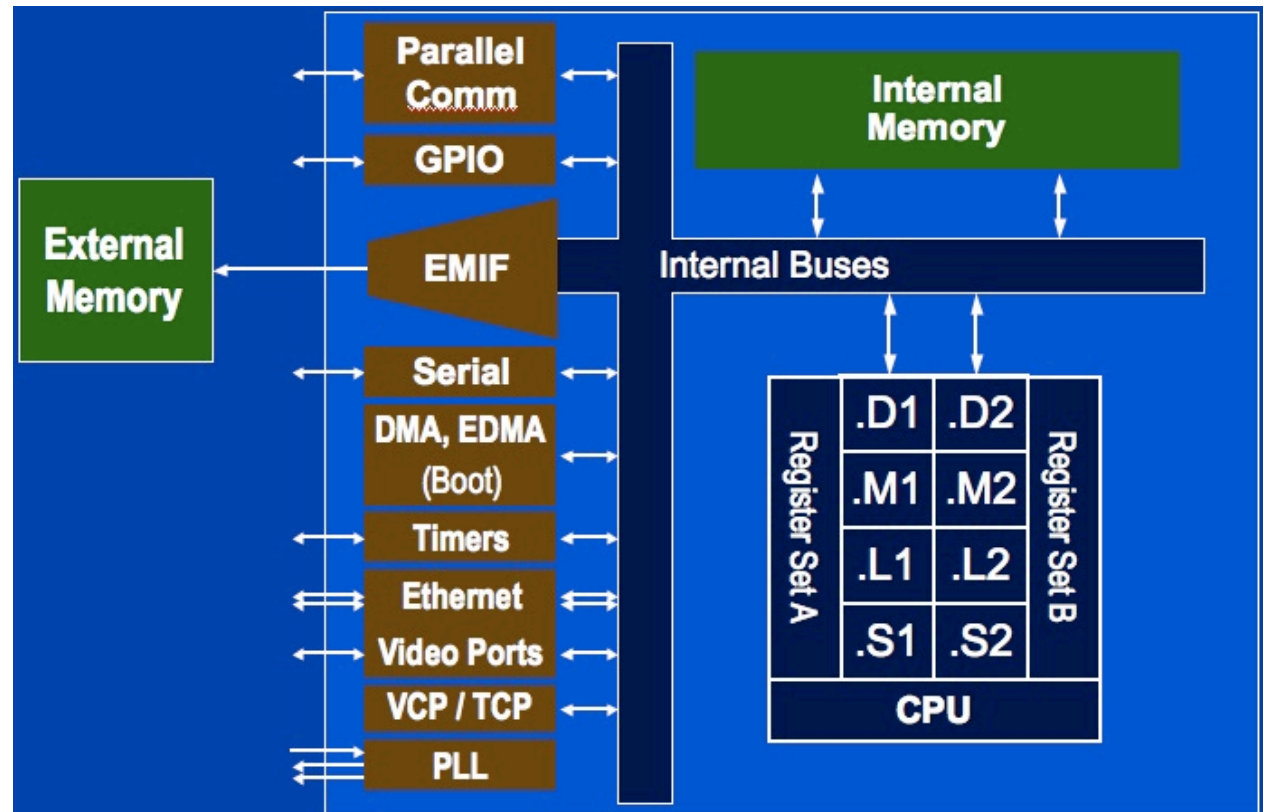
External Memory Interface (EMIF)

- A special hardware interface is required when interfacing a slow peripheral/memory to a fast DSP
- EMIF composed of EMIFA (64-bit) and EMIFB (16-bit)
 - Works with PC100/133 SDRAM (cheap, fast, and easy!)
 - Byte-wide data access 16, 32, or 64-bit bus widths

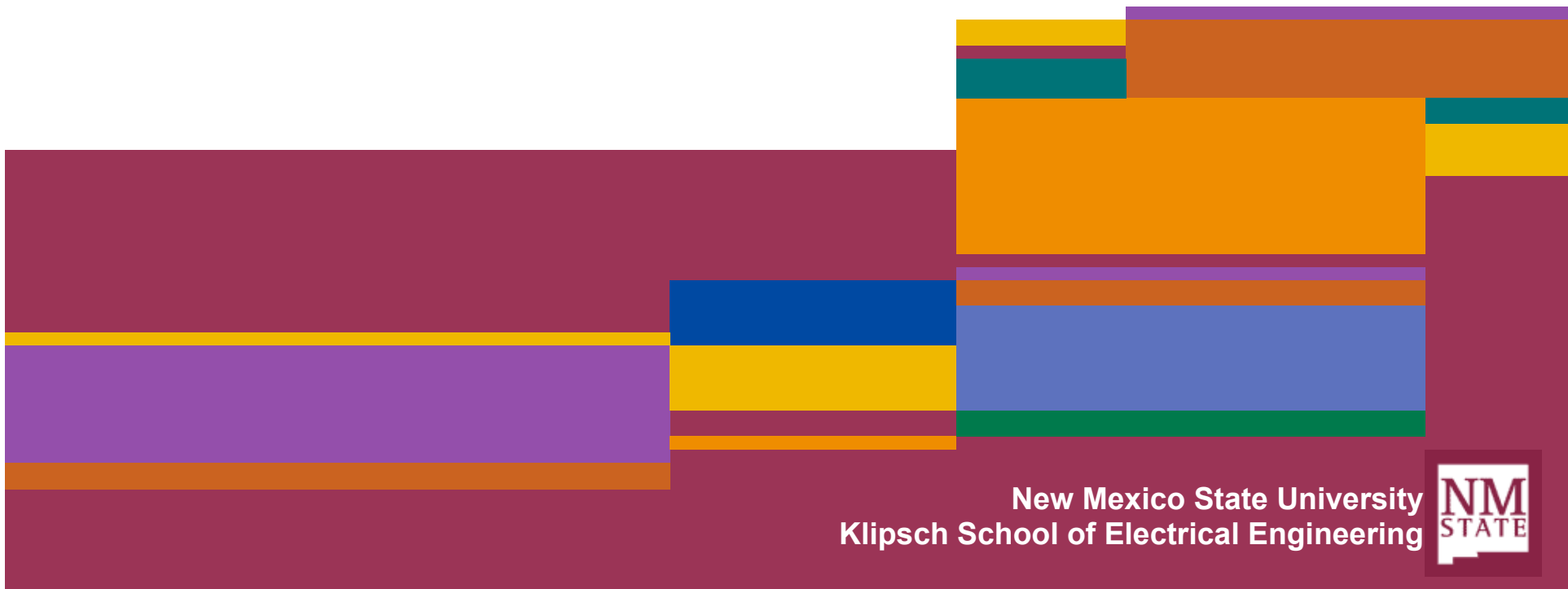


Multi-Channel Buffered Serial Port (McBSP)

- **McBSP**
 - 2 (or 3) full-duplex, synchronous serial-ports
 - Up to 100 Mb/sec performance
 - Supports multi-channel operation (T1, E1, MVIP, ...)



TI C6416DSK (DSP Starter Kit)



New Mexico State University
Klipsch School of Electrical Engineering



Code Composer Studio

- 6416DSK includes an excellent Integrated Development Environment (IDE) called Code Composer Studio (CCS)
 - IDE includes editor, C/C++ compiler, and debugger in a single package
 - CCS used for developing code for TI TMS320 family of DSP chips
 - CCS includes a mini operating system called DSP/BIOS
- DSK also include a pass code called “dsk_app”
 - Prof. DeLeon has significantly modified dsk_app to allow smoother transition from Freescale DSP56302EVM and to integrate with DSP Fixed-Point C-functions

TI AIC23 Codec

- **6416DSK Audio I/O**
 - Inputs: Stereo Line In, Mic In
 - Outputs: Stereo Line Out, Headphone Out
 - All audio flows through McBSP data channel
- **McBSP1: Unidirectional Control Channel**
 - Configures Codec's internal configuration registers
- **McBSP2: Bidirectional Data Channel**
 - Transmits and receives Input and Output audio samples from 6416 DSP
- **Important Settings:**
 - Audio bypass mode
 - Volume control
 - Sample rates: 8, 32, 44.1, 48, 96 kHz
 - Sample resolution: 16, 20, 24 or 32 bits

Other Stuff

- **Synchronous DRAM (SDRAM)**
 - Two 64 Mb SDRAMs on the 64-bit wide EMIFA
 - The SDRAM is mapped at the beginning of CE0 (address 0x80000000)
- **Flash Memory**
 - 6416DSK has a 512 KB external flash memory as a boot option
 - Connected to CE1 of EMIFB with an 8-bit interface
- **Complex Programmable Logic Device (CPLD)**
 - Software board configuration through registers implemented in CPLD
 - Memory mapped registers (for LED and DIP switches)
- **LED and DIP switches (accessed through CPLD registers)**
 - 4 software accessible LEDs and DIP switches provides user a simple form of input/output

Memory Map for 6416DSK

Address	Generic 6416 Address Space	6416 DSK
0x00000000	Internal Memory	Internal Memory
0x00100000	Reserved Space or Peripheral Regs	Reserved or Peripheral
0x60000000	EMIFB CE0	CPLD
0x64000000	EMIFB CE1	Flash
0x68000000	EMIFB CE2	
0x6C000000	EMIFB CE3	
0x80000000	EMIFA CE0	SDRAM
0x90000000	EMIFA CE1	
0xA0000000	EMIFA CE2	Daughter Card
0xB0000000	EMIFA CE3	

Figure 1-2, Memory Map, C6416 DSK