1 Binary/Hexadecimal Numbers

Binary:	256 128 64 32 16 8 4 2 1
Hexadecimal:	4096 256 16 1

Hint:

$$\begin{aligned} 2AF3_{16} =& (2_{16} \cdot 16^3) + (A_{16} \cdot 16^2) + (F_{16} \cdot 16^1) + (3_{16} \cdot 16^0) \\ =& (2 \cdot 4096) + (10 \cdot 256) + (15 \cdot 16) + (3 \cdot 1) \\ =& 10995 \end{aligned}$$

unsigned: span of $[0,\ 2^N - 1]$

Sign/Magnitude: Most significant bit represents the sign (0 indicates positive). Does not work with ordinary binary addition, 0 has multiple representations. Span of $[-2^{N-1} + 1,\ 2^{N-1} - 1]$

Two's Complement: Most significant bit has weight of 2^{N-1}. To reverse the sign, invert all bits of the number and then add 1. Span of $[-2^{N-1},\ 2^{N-1} - 1]$

2 Binary Equations

Minterm: product involving all of the inputs to the function (e.g. $AB\overline{C}$)

Maxterm: sum involving all of the inputs to the function (e.g. $A + \overline{B} + \overline{C}$)

Sum-of-Products: Take all rows of a truth table that are TRUE, multiply their variables and add them up

Product-of-Sums: Take all rows of a truth table that are FALSE, add the complements of their variables and multiply them with each other

3 Combinational Logic Design

Rule of thumb: The outputs depend strictly only on the input.

- Every circuit element is itself combinational
- Every node of the circuit is either designated as an input to the circuit or connects to exactly one output terminal of a circuit element
- The circuit contains no cyclic paths: every path through the circuit visits each circuit node at most once

Contention: Unknown/illegal value (X). When a node is driven to both, HIGH and LOW. Error, should be avoided. Doesn't have to be in the forbidden zone though.

Floating Value: Floating, high impedance (Z). When a node is driven neither to HIGH nor LOW. Not necessarily an error.

3.1 K-Maps

- Use the fewest circles necessary to cover all 1's
- All the squares in each circle must contain 1's
- Each circle must span a rectangular block that is a power of 2 (i.e., 1, 2, or 4) squares in each direction
- Each circle should be as large as possible
- A circle may wrap around the edges of the K-map

- A 1 in a K-map may be circled multiple times if doing so allows fewer circles to be used
- Don't Cares can be circled if they help cover the 1s with fewer or larger circles, however they don't have to be circled

To get the result:

sum of products: circle the 1's, multiply the variables that don't change for that circle

product of sums: circle the 0's, add the complements of the variables that don't change for that circle

The transition across the boundary of two prime implicants in a K-map indicates a possible **glitch** (flickering between outputs before it settles to the final one).

3.2 Delays

Propagation delay t_{pd}: Maximum time from when an input changes until the output or outputs reach their final value

Contamination delay t_{cd}: Minimum time from when an input changes until any output starts to change its value

4 Sequential Logic Design

Depends on current input and prior inputs (has memory, hence state).

4.1 Memory

Multistable elements with N stable states carry $\log_2(N)$ bits of information.

SR-Latch: Two cross-coupled NOR gates with two inputs: S (sets output to 1) and R (resets output to 0). Odd, as one can assert S and R simultaneously.

D-Latch: Clock input which makes the latch **transparent** (data from D flows to Q) on 1. Is CLK unasserted, the latch is **opaque** (remembers the previous D input).

D Flip-Flop: Two back-to-back D latches with complementary clocks. Copies D to Q only on the rising edge of CLK and remembers its state at all other times.

Register: N-bit register is a bank of N flip-flops with a common CLK input

Enabled Flip-Flop: additional EN input, ignores the clock if EN is not asserted

Resettable Flip-Flop: additional RESET input, resets the output to 0 if asserted

4.2 Finite State Machines

Moore Machines: output depends only on the current state

Mealy Machines: output depends on both, current state and input

4.2.1 State Encoding

Goal is to determine the encoding that produces the circuit with the fewest logic gates or the shortest propagation delay.

One-hot: one bit per state, simple, few gates

5 Arithmetic

5.1 Number Systems

fixed-point: implied decimal point, columns behind that point represent 2^{-1}, 2^{-2} and so on

floating-point: analogous to scientific notation, decimal point floats to the position right after the most significant digit. The mantissa only stores the fraction bits, as the leading bit is always 1 (implicit leading one). To represent negative exponents, biased exponents are used (e.g. the exponent 7 is represented as $7 + 127 = 134 = 10000110_2$). 32 bits are used to represent 1 sign bit, 8 exponent bits, 23 mantissa bits and uses a bias of 127.

6 Memory

6.1 RAM

DRAM: Dynamic RAM, cheap, each bit of data in a separate capacitor, needs to be refreshed every few milliseconds, used for main memory (RAM)

SRAM: Static RAM, more expensive, faster, does not need to be refreshed, used for cache of CPU

6.2 Locality

Temporal Locality: keep recently accessed data in higher levels of memory hierarchy

Spatial Locality: when accessing data, bring nearby data into higher levels of memory hierarchy too

6.3 Performance

$$AMAT = t_{cache} + MR_{cache}(t_{MM} + MR_{MM}(t_{VM}))$$

AMAT: Average memory access time

MR: Miss rate (# misses/ # memory accesses)

MM: Main memory

VM: Virtual memory

7 Cache

Capacity (C): number of data bytes a cache stores

Block size (b): bytes of data brought into cache at once

Degree of Associativity (N): number of blocks in a set

Number of sets (S): each memory address maps to exactly one cache set

Categories Can be categorized by number of blocks in a set:

direct mapped: 1 block per set (addresses with one word difference (32 bit) map to the same set)

n-way set associative: N blocks per set

fully associative: all blocks in a single set

Hint: Bigger caches reduce capacity misses, greater associativity reduces conflict misses. Bigger block size reduces compulsory misses but increases conflict misses.

Misses

Compulsory: first time data is accessed
Capacity: cache too small to hold all data of interest
Conflict: data of interest maps to same location in cache

8 MIPS

8.1 Jumping

to a register: **jr $ra**, use this to jump to the calling subroutine
unconditionally: **j** [label]
to a subroutine: **jal** [label] with **$a0 - $a3** argument values and **$v0** as a return value

The **callee** (called method) must preserve the registers **$s0 - $s7**. Likewise, the **caller** must save the registers **$t0 - $t9**, as they may be overwritten.

8.2 Loading

address: **addi $t0, $0, 0x400** to set the register to the address 0x0000 0400
load: **lw $t4, 4($a0)** with 4 as an integer offset
lw $t4, 0xff80($a0) to address **0xFFFF FF80** (sign extension)
byte addressing: **sll $t2, $t1 2** multiplies by 4 (as words are 32 bits)

8.3 Stack

To store 3 registers on the stack on retrieve them again:

```
addi $sp, $sp, -12
sw $s0, 8($sp)
sw $t0, 4($sp)
sw $t1, 0($sp)
```

call subroutines and stuff that potentially overwrite the registers

```
lw $t1, 0($sp)
lw $t0, 4($sp)
lw $s0, 8($sp)
addi $sp, $sp, 12
```

8.4 Loading 32 bit Immediates

To load the address **0x0123abcd** into **$t0**:

```
lui $t0, 0x0123
ori $t0, $t0, 0xabcd
```

9 Verilog

- **Order of the pins in an instantiation** of a module incorrect (outputs connected to the inputs)
- Pins on the left side of an assignment in an **always block** not declared as **reg**
- **Case/If** statement not in the always block