# Welcome back to CS529H!

Week 5

# Old Piazza post of the week:

Actions

# Student generated spec, good idea? bad idea?

Should we do more of it? What adjustments need to be made? ....

logistics hw2

good note | 0

Updated 1 year ago by Ahmed Gheith

**followup discussions** *for lingering questions and comments*

**4 endorsed followup comments**

Resolved Unresolved @177_f1

**Anonymous Atom** 1 year ago

:/

helpful! | 3

**Ahmed Gheith** 1 year ago

**:/** is an emoticon used to indicate indecision, skepticism, exasperation, and annoyance.

**~ An instructor (Nikita Sharma) thinks this is a good comment ~**

undo good comment | 9

# Ed meme recap:

t5.s #253

Michael Goppert
2 days ago in General

PIN STAR WATCH 149 VIEWS

4

CMON... DONT GENERATE 5000 LINES... ITS JUST A CONSTANT EXPRESSION

MY T5 ASM

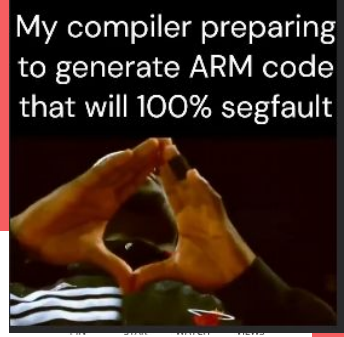
Annie Hu 2d
im x86 fangirl

donate bits to me and ill post x86 content <3

Reply Edit Delete

Michael Goppert 1d
wtf go get hobbies dude.

3 Reply




q1 #248

Anonymous
2 days ago in General

7

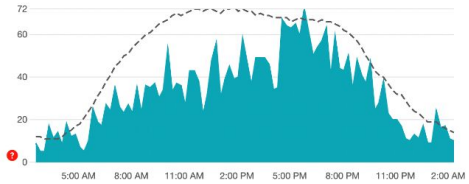
Womp Womp #235

Anonymous
4 days ago in Assignments – P4

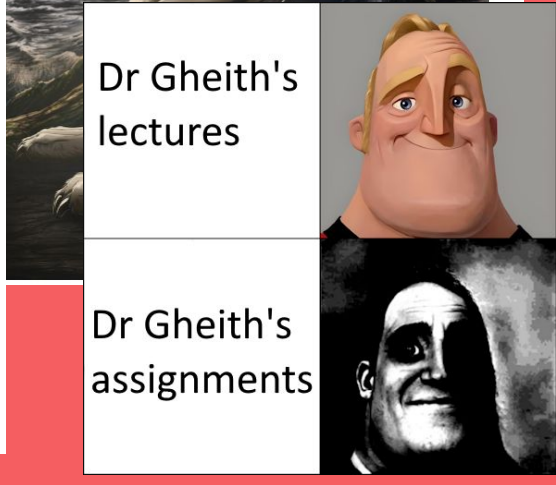
PIN STAR WATCH 229 VIEWS

18

Matrix outages reported in the last 24 hours



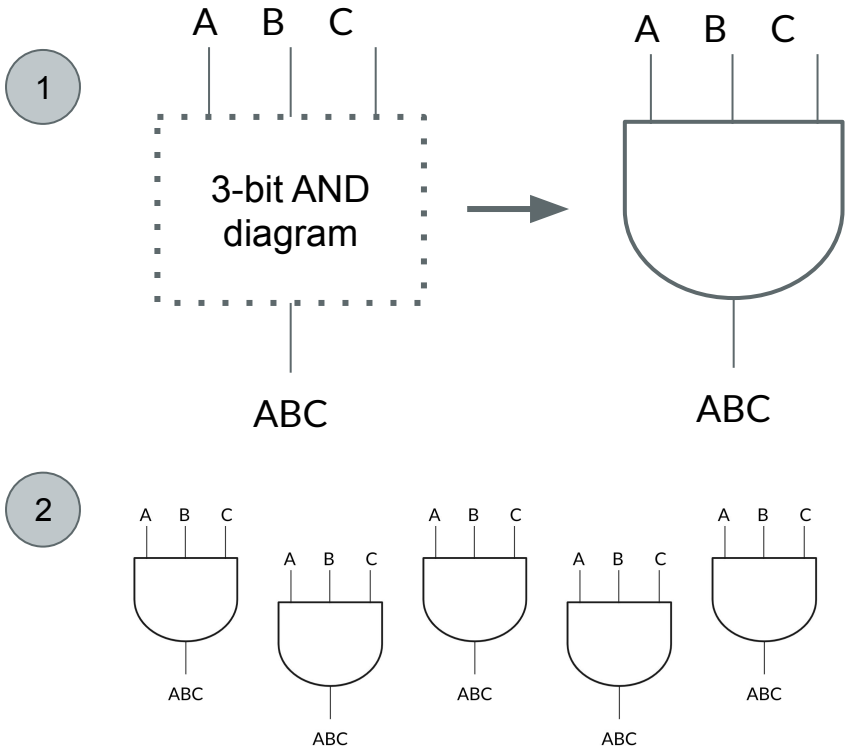

This chart shows a view of problem reports submitted in the past 24 hours compared to the typical volume of reports by time of day. It is common for some problems to be reported throughout the day. Downdetector only reports an incident when the number of problem reports is significantly higher than the typical volume for that time of day. Visit the Downdetector Methodology page to learn more about how Downdetector collects status information and detects problems.



Questions on lecture content?
Or about cats?

# Circuits on the quiz

If you want to use something more complex than the building blocks we allow in a question, show how to implement it once, and then use that.



# Note on Quizzes

- Sorry for making quiz 1 so hard!!
- How should we do quiz clarifications?

Quiz everyone say YIPPEE!

# Poll

```
 adrp x0, :pg_hi21:feedback
ldr x0, [x0, :lo12:feedback]
```

How was the quiz?

A. easy
B. mostly fine
C. mostly fine, but not enough time
D. too hard, but finished mostly in time
E. too hard and not enough time
F. too hard regardless of time

# Stress

- 429H is not an easy class
  - Lots of new materials
  - Unfamiliar programming environments
  - Fast, often relentless pace
- Struggling in this course is normal
  - There will be times you won't know the answer of the solution
  - This is expected—we want we everyone to succeed, but the only way we can help is if you ask for it
- If you find yourself overly overwhelmed or spending more time on this class than you think you should be, please reach out to Dr. Gheith or the TAs
  - We can help out as far as the class goes
  - We can provide other resources where we are not able to help

Mental health resource available at UT

# Why are we on p4 for the third discussion

# Poll

How’s your status on P4?

A. What’s P4?
B. I’ve heard of it
C. I’ve cloned the starter code and/or looked through it
D. I’ve started planning/writing code
E. I’m mostly done but might still have bugs
F. P4 any% speedrun

# Constant folding

- Pre-evaluating constants at compile time to simplify necessary computation
- `1 + 2 + 3 + x -> 6 + x`
- How do you **interpret** part of an expression to simplify it?
- Bonus question: can we be sure that constant folding won't change the output?
  - Ask Alex or Caleb about this one (or take PL next year)
- Stronger version: Just interpret everything that isn't dependent on `argc`
  - Requires multiple passthroughs and probably not worth it at this point
  - Requires more complicated data structures (how to track dependencies?)

# Tail recursion optimization

- Preferred paradigm especially in functional programming languages
- Used by real compilers to prevent stack overflows
- Only works when the recursive call is the final statement
  - Good: `return f(x)`
  - Bad: `return 5 + f(x)`
- How to implement in our compiler?
  - Reuse stack frame
  - Does it have to be recursive?

# </p4>

(unless you still have questions at the end)

# How do we represent an arbitrary truth table as a circuit?

| A | B | C | D | out |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# Karnaugh Maps

| CD \ AB | *00* | *01* | *11* | *10* |
|---|---|---|---|---|
| *00* | 0 | 0 | 1 | 1 |
| *01* | 0 | 1 | 0 | 1 |
| *11* | 1 | 1 | 1 | 1 |
| *10* | 1 | 1 | 1 | 1 |

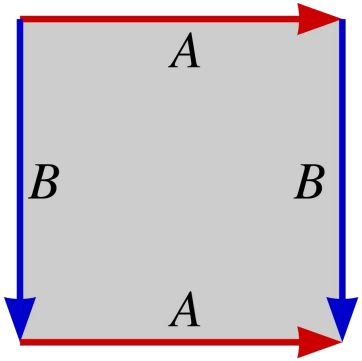
# C + AB + AD + $\bar{A}BCD$

| CD \ AB | *00* | *01* | *11* | *10* |
|---|---|---|---|---|
| *00* | 0 | 0 | 1 | 1 |
| *01* | 0 | 1 | 0 | 1 |
| *11* | 1 | 1 | 1 | 1 |
| *10* | 1 | 1 | 1 | 1 |

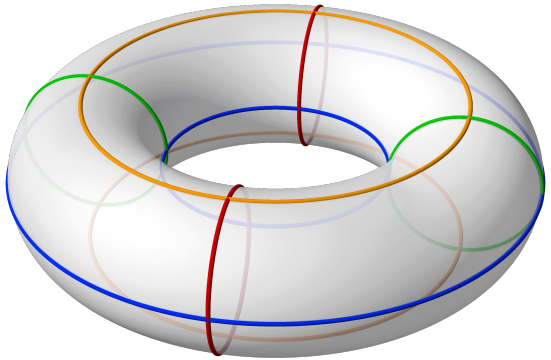
# Fun fact

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

≃

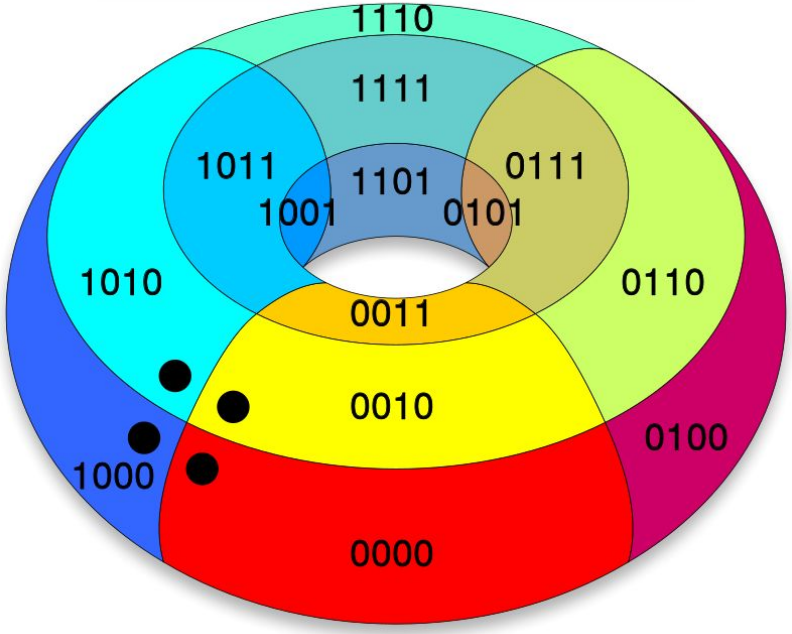



≃



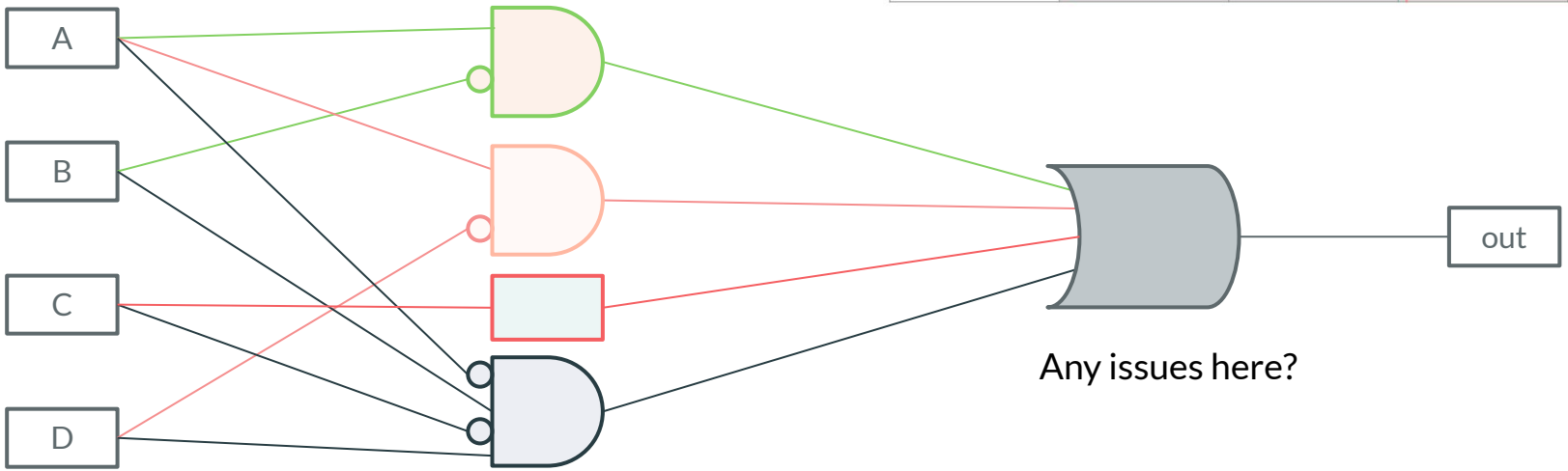
≃


0000
0100
1100
1000
0001
0101
1101
1001
0011
0111
1111
1011
0010
0110
1110
1010


1110
1111
1011
1101
0111
1001
0101
1010
0110
0011
0010
0100
1000
0000

# C + AB + AD + $\bar{A}BCD$

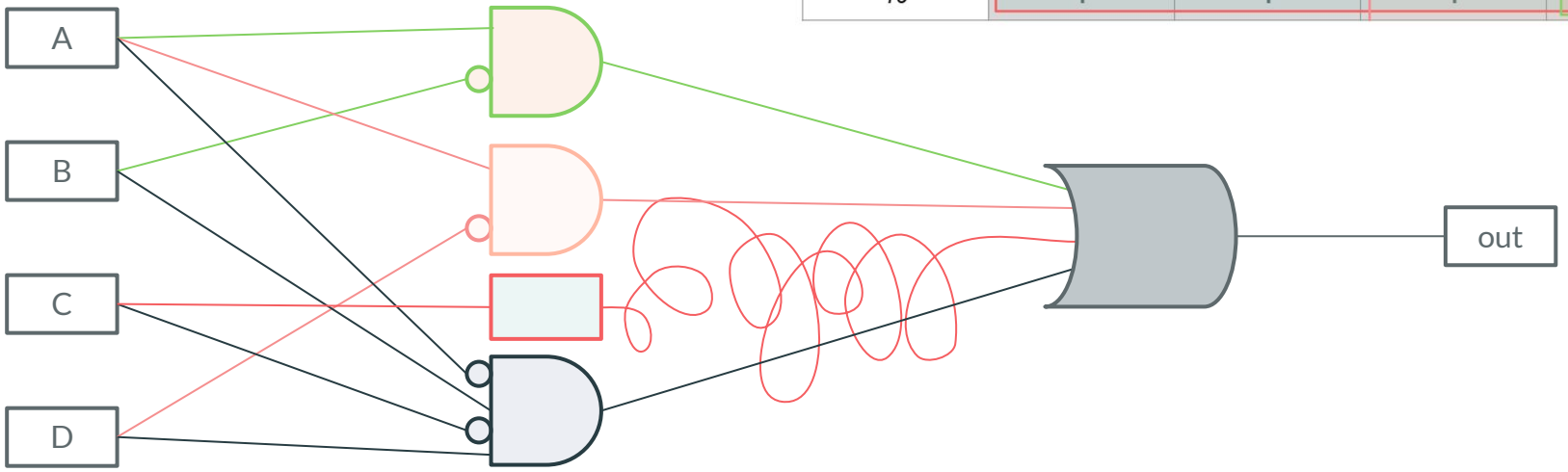
| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |




Any issues here?

# C + AB + AD + ĀBCD

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

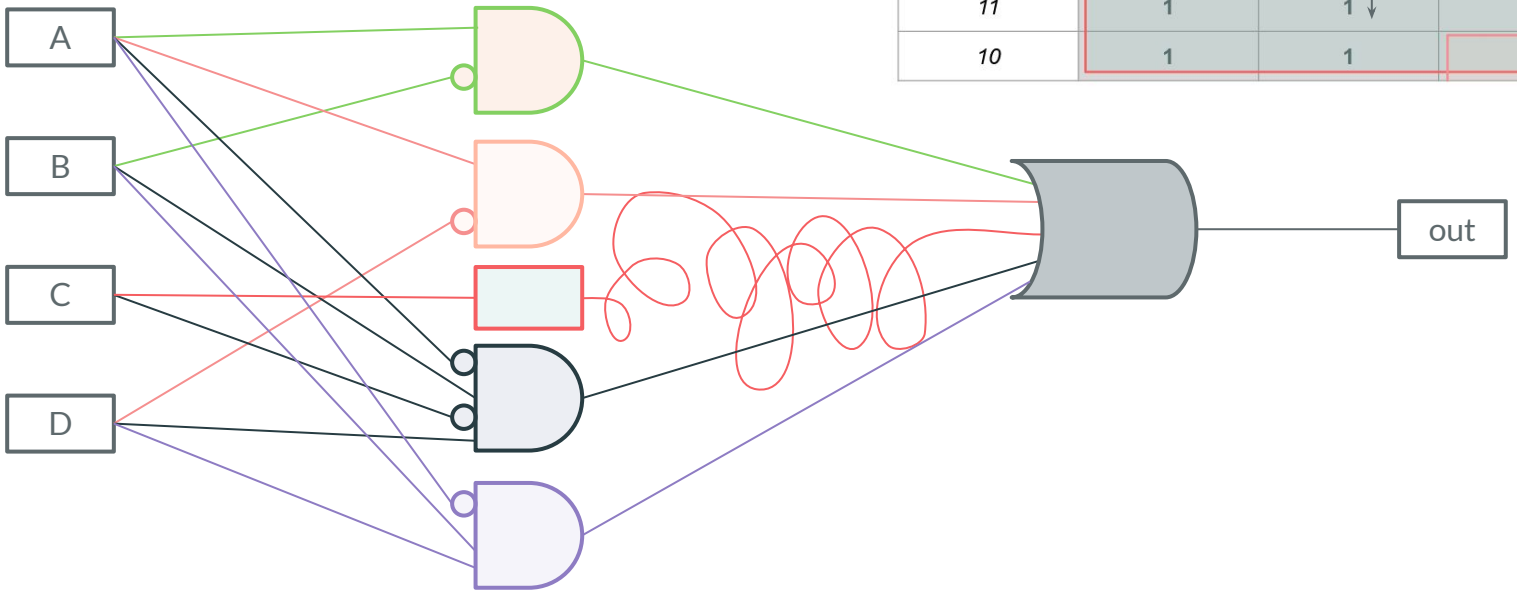


# C + AB + AD + $\bar{A}$BCD + $\bar{A}$BD

| CD \ AB | *00* | *01* | *11* | *10* |
|---|---|---|---|---|
| *00* | 0 | 0 | 1 | 1 |
| *01* | 0 | 1 | 0 | 1 |
| *11* | 1 | 1 | 1 | 1 |
| *10* | 1 | 1 | 1 | 1 |

# C + AB + AD + ĀBCD + ĀBD

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |



# Poll

Should we do a fun activity next week?

A. Yes
B. Yes

P5

# Poll

How's your status on P5?

A. What's P5?
B. I've heard of it
C. I've cloned the starter code and/or looked through it
D. I've started planning/writing code
E. I'm mostly done but might still have bugs
F. P5 any% speedrun

# How to heap 2: electric boogaloo

(this is mostly just a review from last week in case you forgor)

(if we even get to cover these slides because quiz is long)

# How to Heap

- Heap is hard
- *Consistency*
- Keep consistency through *invariants*
  - An *invariant* should be true at the **beginning** and **end** of all heap functions
  - They can be violated *temporarily* in the middle of these functions
  - Examples of invariants?

# How to Heap

- Structs and Functions! Strunctions!
- What kind of structs might you need?
- What kind of functions might you need?
- What kind of strunctions might you need?

# How to Heap

```
struct __attribute__((packed)) foo {
    int a : 2;
    int b : 6;
};
```

- What is sizeof(struct foo)?
- Why might you want to do this?

# Debugging Tips

- How can you check your invariants?
- Diagnostics—use a heap_check() function
  - Pretty-print entire heap state
  - Check invariants programmatically
  - Call after every malloc/free
    - + Catch bugs early
    - − Makes your code slooow ⌛
  - Call only in certain cases
  - e.g. (gdb) call print_heap()
    - + Less verbose / spammy
    - − Lower coverage
- Downsize the test case
  - Small test cases are easier to debug
- Debug interactively with gdb!
  - watch and rwatch
  - (gdb) watch head
  - (gdb) rwatch (long *) 0x832a8b0
  - (gdb) watch curr_block->free

# Debugging Tips

```
CFLAGS = -Werror -Wall -O3 -g -std=c11
```

changes to

```
CFLAGS = -Werror -Wall -O0 -g -std=c11
```

**Why can’t we use printf debugging for the heap?**

# Debugging Tips - Conditional Compilation

```
CFLAGS = -Werror -Wall -O3 -g -std=c11
```

changes to

```
CFLAGS = -Werror -Wall -O3 -g -std=c11 -DDEBUG
```

Then in your code, you can have

```
#ifdef DEBUG
heap_check()
#endif
```

# Questions?