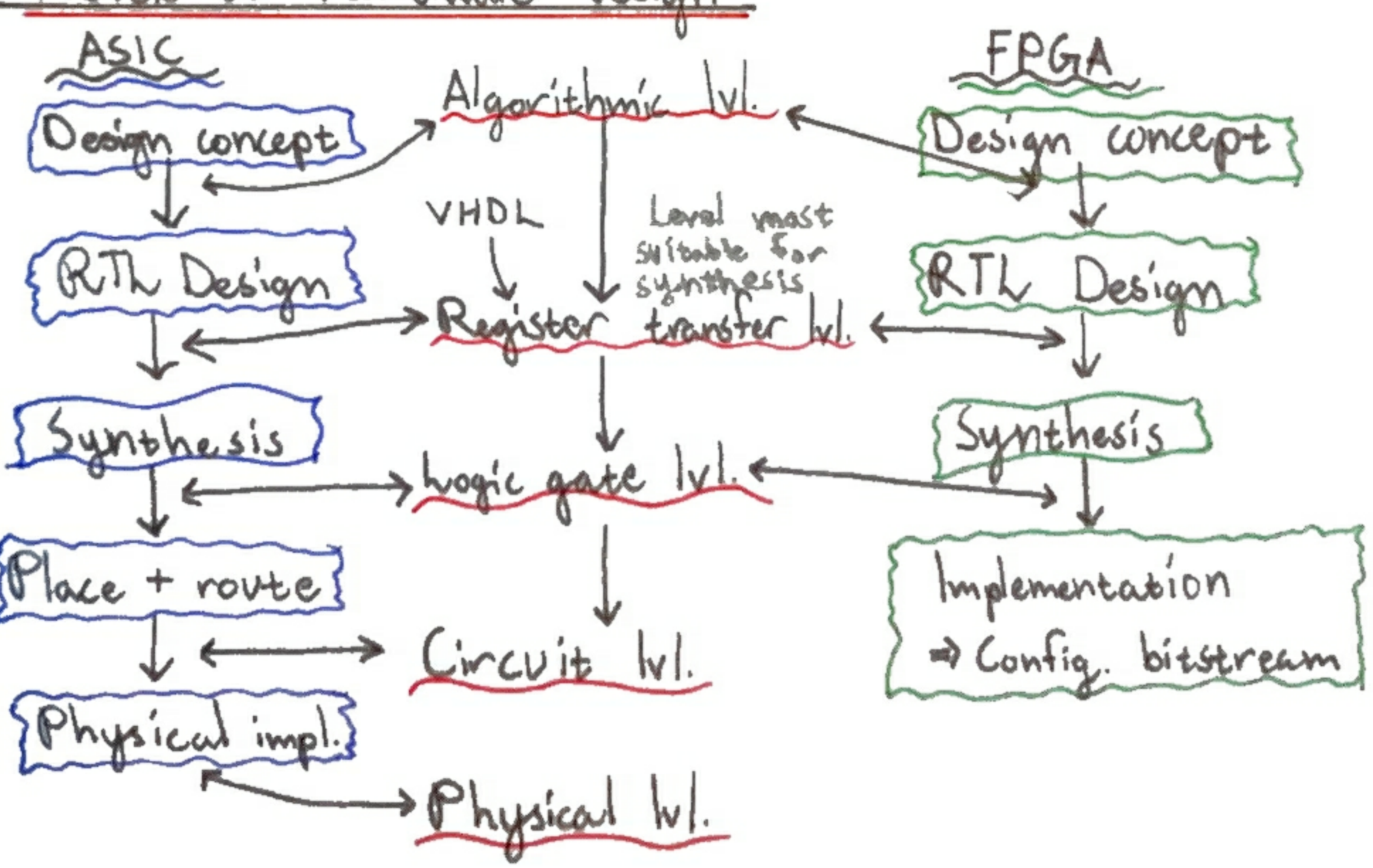
Digital design, summary, week 1

- ASIC - Application specific integrated circuits
  - Cannot change
  - All the way from behavorial description to physical layout
  - Expensive & time consuming fabrication
  - High performance, low cost in high volumes

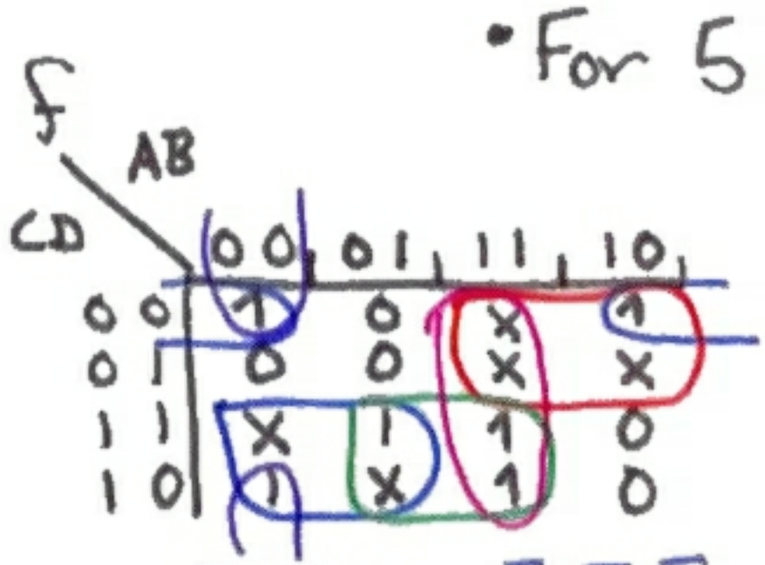
- FPGA - Field programmable gate arrays
  - Generic - support different hardware designs
  - Behavorial description → Bitstream → loaded to FPGA
  - Bought off the shelf
  - Low development cost

- Levels of hardware design



- VHDL Design styles
  - Dataflow - Boolean functions, parallell
  - Structural - Components & interconnects
  - Behavorial - More like software, sequential
- Bodean algebra
  - $\overline{X \cdot Y} = \overline{X} + \overline{Y}$
  - $\overline{X + Y} = \overline{X} \cdot \overline{Y}$
  - $X + \overline{X} \cdot Y = X + Y$

  *
  - Associative: $(X \cdot Y) \cdot (Z) = (X) \cdot (Y \cdot Z)$
  - Distributive: $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
  - Commutative: $X + Y = Y + X$
- De Morgan *
  - AND-OR $\Leftrightarrow$ NAND-NAND
  - OR-AND $\Leftrightarrow$ NOR-NOR
- Minterm - all variables appear once, product term. $A \cdot B \cdot \overline{C}$
- Maxterm - sum term instead. $(A + B + \overline{C})$
- Canonical form - all terms are miterms / all are maxterms
- $F = \sum$ minterms $= \prod$ (sums) maxterms, $F' = \prod$ minterms $= \sum$ max
- Karnaugh-diagram
  - For simplification
  - 1's - minterms - S.P
  - 0's - maxterms - P.S
  - For 5 variables - do two



All possible:

$F = A\overline{C} + \overline{A} \cdot C + BC + A \cdot B + \overline{A} \cdot \overline{B} \cdot \overline{D} + \overline{B} \cdot \overline{C} \cdot \overline{D}$

Simplest: $F = B \cdot C + A \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot \overline{D}$, for example

- Definition of terms
  - A function f covers a function g if f='1' when g='1'
  - Implicant - Aprodvct of variables for which f='1'
  - Prime implicant - An implicant that can't be covered by a more general expression
  - Essential prime implicant - A prime implicant that includes a minterm which is not included by any other prime implicant.
- Edges doesn't line up exactly in simulation because it takes time for gates to change it's output.
- Delay - Nr. of gates in a row
- Tradeoffs: • More area for lower delay
  • Higher delay for lower power
- Critical path - Longest path from input to output
- Ripple carry adder - several full adders
- A - B = A + B_{2c}. Invert all bits and add one.
- Carry select adder - keep second part of nr. in two versions (C_{IN} = 0 and C_{IN} = 1)
- Carry look-ahead adders
  - A carry can either be generated or propagate thrugh
  - Generated if A = 1 & B = 1. $g_i = a_i \cdot b_i$
  - Propagates if either A or B = 1 $\Rightarrow P_i = a_i \oplus b_i$
  - $C_{IN\,i+1} = g_i + P_i \cdot C_{IN\,i}$, $S_i = P_i \oplus C_{IN\,i} = (a_i \cdot b_i) \oplus C_{IN\,i}$
- Gives us logarithmic time complexity

• Chalmers accumulator processor - ChAcc

- Accumulator - keeps result of most recent operation
- Simple but slow - ~2-4 clock cycles per instruction
- Operates on 8-bit data
- Datapath - where data flow
  - Used / modified
  - Read / written from / to memory
  - Memory, registers, ALU
- Controller - synchronizes processor's components and orchestrates operations
- Communications bus - Transfers data
- Harvard architecture - separate memory for instructions and data. 8-bit addresses.
- 4-bit op code and 8-bit argument
- Controller breaks datapath into five stages
  - Fetch, decode, decode*, execute, memory
  - Instructions doesn't pass through all stages
- Controller is a finite state machine