Mobile Systems

Lecture 10 – 21/4/15

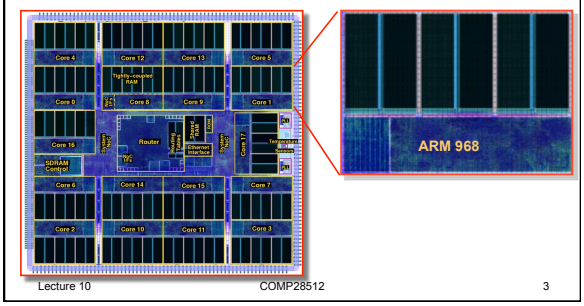
Chip design

COMP28512

Steve Furber & Barry Cheetham

Baby (1948)

SpiNNaker CPU (2011)



63 years of progress

- ***Baby:***
  - used 3.5 kW of electrical power
  - executed 700 instructions per second
  - 5 Joules per instruction
- ***SpiNNaker ARM968 CPU node:***
  - uses 40 mW of electrical power
  - executes 200,000,000 instructions per second
  - 0.000 000 000 2 Joules per instruction

25,000,000,000 times better than Baby!

(James Prescott Joule born Salford, 1818)

Jevons paradox

1865 "The Coal Question"

- James Watt's coal-fired steam engine was much more efficient than Thomas Newcomen's…
- …and coal consumption rose as a result

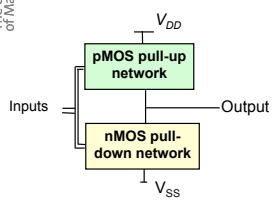
Low-power design

- Why design for low power consumption?
  - mobile: require maximum battery life
  - HPC: exascale needs extreme efficiency!
- Better power-efficiency
  - more performance/functionality
  - more CPU cycles for compression and ECC
  - to reduce radio power
  - better battery life
- Lower power gives market advantage

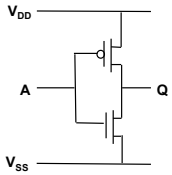
CMOS power consumption

- CMOS power consumption
  - voltage change on a gate capacitance requires *charge transfer*, & therefore power consumption
  - once a gate is charged it can maintain its level without any additional charge movement
- CMOS circuitry **only** consumes power when switching states
  - well, until leakage starts to bite!

CMOS circuits



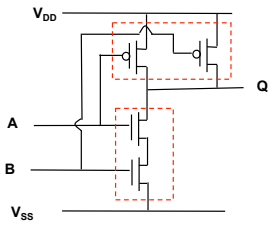
In general



An inverter

- Can make Vss = 0 (GND).
- When not switching (static) only pull-up or pull-down network conducts.
- Not both.

CMOS NAND-gate



Dynamic power consumption

$$P = 1/2 \times f_{clock} \times V_{DD}^2 \times \sum_{\text{all gates}} \alpha_g C_g \quad \text{Watts}$$

where:

f_{clock} = switching frequency of device clock

V_{DD} = supply voltage (assuming V_{SS}=0)

C_g = capacitance load on gate g

α_g = 'activity' on gate g:
= mean number of transitions per clock cycle
= 2 for a clock signal, ≈ 0.1 otherwise

Dynamic power consumption (simplified expression)

$$P = 1/2 \text{ X } C_{total} \text{ X } f_{clock} \text{ X } V_{DD}^2 \text{ X } \alpha \quad \text{(Watts)}$$

where:

C_{total} = total node capacitance

f_{clock} = switching frequency of device clock

V_{DD} = supply voltage

α = mean overall activity:
= mean number of transitions per clock cycle
= 2 for gates connected to a clock

Energy consumption

- If P is constant, energy = E = P x time (Joules)
- In battery powered mobile devices, we have to worry about both P and E.
- If P is too high, even for a short time, heat may cause damage to circuitry.
- If P is reduced, but computing time is too high, battery life will be limited.
- Can run intermittently by switching power on/off
  - maybe to different parts of a circuit
  - good way to save power & energy
  - but 'just in time' processing may be better?
  - tortoise & hare problem!

MANCHESTER 1824

The University of Manchester

Reducing dy power consumption 1

$$P = {}^1/_2 \times C_{total} \times f_{clock} \times V_{DD}^2 \times \alpha \quad \text{(Watts)}$$

- Reducing V_{DD} greatly reduces P
- But it also decreases the current than can be supplied by each transistor when it is switched on.
- Lower supply voltage means lower current.
  - Load capacitances will charge more slowly.
  - Gate switching will become slower
  - Maximum possible value of f_{clock} will reduce
  - Programs may take longer to run
- V_{DD} used to be 5 volts (in my day).
  - Now down to ≈ 1.2 v
- Threshold voltages (Vth) for transistors use to be 0.7 v.
  - Now down to ≈ 0.2 v
- Can use parallelism to offset increases in circuit delay.

MANCHESTER 1824

The University of Manchester

Theshold voltage (Vth) ?

- Voltage that must be exceeded for a transistor to switch from 'off' to 'on'.
- Depends on many factors determined by manufacturer
  - e.g. doping level
- Clearly, V_{DD} must be greater than Vth for all transistors.
- It may be shown that:

$$f_{clock}(max) \propto (V_{DD} - Vth)^2 / V_{DD}$$

- So, to reduce V_{DD} without slowing down the circuit, why not reduce Vth ?
- Unfortunately there is another problem:

  Leakage current $\propto \exp(-Vth / 0.035) \approx 10^{-Vth/0.08}$

  Will cause battery drain through an inactive circuit

  We begin to lose advantages of CMOS
- Can mix low & high values of Vth.

MANCHESTER 1824

The University of Manchester

Reducing dy power consumption 2

$$P = 1/2 \times C_{total} \times f_{clock} \times V_{DD}^2 \times \alpha$$

- Reducing f_{clock} also reduces P
- But consider effect on energy for running a given program
  - time to complete computation $\propto 1 / f_{clock}$
  - power $\propto f_{clock}$
  - so energy to run a program remains the same
  - number of instructions per Joule independent of f_{clock}
  - reducing f_{clock} is only a good idea if it allows lower V_{DD}

MANCHESTER 1824

The University of Manchester

Reducing dy power consumption 3

$$P = 1/2 \times C_{total} \times f_{clock} \times V_{DD}^2 \times \alpha$$

- Reducing C_{total} will clearly reduce P
- How can we do this?
  - use smaller, simpler circuits
    - e.g. ARM core rather than Pentium
  - do not over-size gates and buffers
    - in particular, reduce drive off critical path
  - use on-chip rather than off-chip memories
    - off-chip capacitances >> on-chip

MANCHESTER 1824

The University of Manchester

Reducing dy power consumption 4

$$P = 1/2 \times C_{total} \times f_{clock} \times V_{DD}^2 \times \alpha$$

How to reduce activity factor(s) α ?

- design circuits that do not switch more than is necessary
- use gates to avoid unnecessary distribution of clock signals
- turn off processor when it has nothing to do
  - don't make it sit in an idle loop!
- use an 'event-driven' style of design
  - in the limit, use asynchronous design (globally or locally)

MANCHESTER 1824

The University of Manchester

Example: SpiNNaker

- Power budget
  - ARM968 (from ARM web site)
    - 0.12 - 0.23 mW/MHz on 130 nm CMOS
    - 24 - 46 mW at 200 MHz
  - 20 x ARM968 = 480 – 920 mW (flat out!)
  - Comms link
    - 1.8 V I/Os, 10 pF/wire = 16 pJ/transition
    - 18 active outputs at 250 MHz = 70 mW (max)
  - Router – 200 mW (guess – check!)
  - SDRAM controller – 50 mW ??
  - Total: 800 – 1,240 mW (max)
    - aim to keep under 1 watt for low-cost packaging

Leakage (again)

- Transistor off current is not zero!

$$I_{off} \propto 10^{(-V_t/100mV)}$$

  - V_t is the transistor threshold
- In my day, when V_{DD} = 5 V, V_t = 0.7 V, I_{off} ~ pA
  - x 1,000,000 transistors = 1 μA (not much to worry about)
- In deep submicron CMOS V_{dd} is lower
  - e.g. 130 nm, V_{DD} = 1.2 V, V_t = 0.3 V, I_{off} ~ 10 nA
  - x 100,000,000 transistors = 1 A
- This is a big problem!
  - leads to unacceptable standby power in mobile systems

Chip design

- Mobile systems
  - are complex SoCs
  - all functions are on a single chip
    - except perhaps large memories, RF amps
  - based around re-usable "IP blocks"
    - some bought in (e.g. ARM processor cores)
    - some proprietary
  - interconnected using bus architectures
  - all built using automated design tools
    - which still require a lot of skill and time to drive!

Chip design

- Principles of low-power design
  - minimize activity
    - but trade-off compression/ECC/RF power
  - localize activity
    - use on-chip rather than off-chip memory
    - use cache to minimize access to large off-chip memory
    - plan streaming data movement to minimize bandwidth and distance moved

RISC architecture

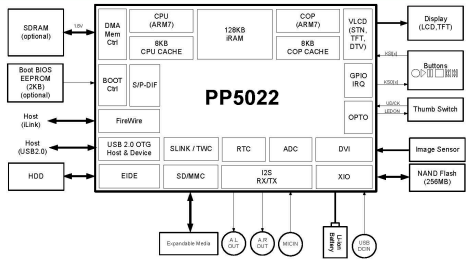
•According to SBF's book, what are processors doing most of the time?

Instruction type	Dynamic usage
Data moves	43 %
Control flow	23 %
Arithmetic	15 %
Comparisons	13 %
Logical ops	5 %
Others	1 %

← Note

- RISC introduced simplicity e.g. by having hard-wired instructions
- Pipelined execution optimised for the table above
- Many single cycle instructions. For the full story, see SBF's book.

iPod hardware



Mobile Systems

- The future:
  - Moore's Law
    - still has a little way to go
      - more functionality/power
      - escalating cost per design
      - and beware of leakage!
  - digital media
    - speech, music, image, video
      - big improvements unlikely?
  - digital communications
    - mobile bandwidth still increasing
  - product diversity
    - what will be the next big consumer market for mobile systems?