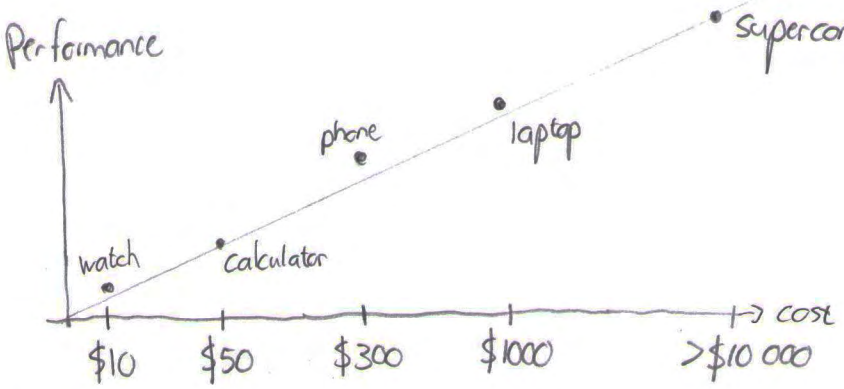
TOTAL:
CS150/EE141 = 50 pts
EE241 = 60 pts

Homework 1 : Solutions

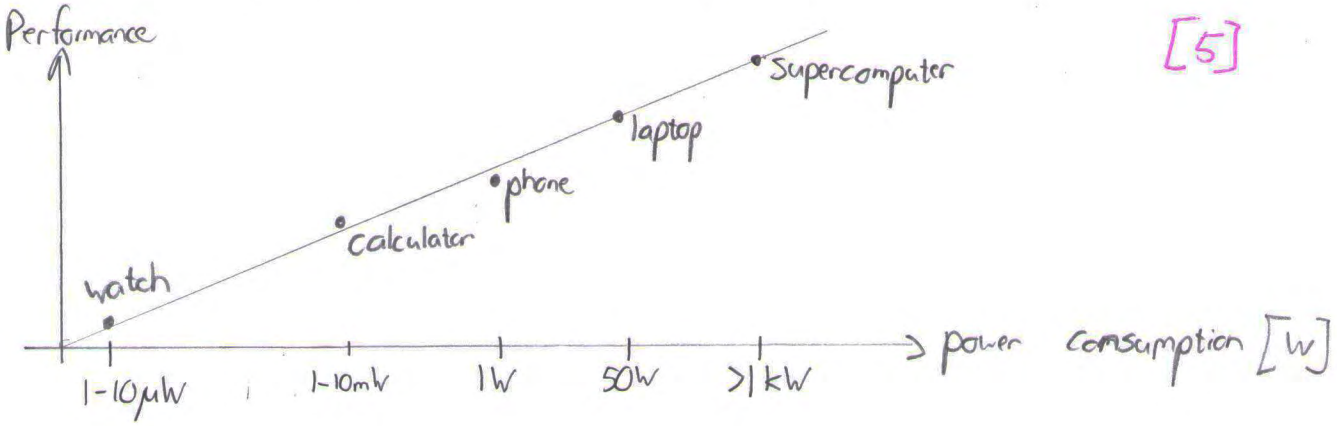
Q1

a) [5]



Note: exact cost estimates not very important

b.) [5]



NB: for both cases, the axes are not scaled. Therefore, cannot determine if slope is linear, quadratic, etc.

Q2

a.) [2]

$T_A = \frac{5000}{1} \times \frac{1}{2\times10^9} = \underline{2.5\,\mu s}$

$T_B = \frac{5000}{2} \times \frac{1}{1\times10^9} = \underline{2.5\,\mu s}$

b.) [4]

$P_A = P_{transistor} \times N_{transistor}$

$= 2\times10^{-14} \times 1.2^2 \times 2\times10^9 \times 2\times10^6$

$= \underline{115.2\,W}$

$P_B = 2\times10^{-14} \times 0.7^2 \times 1\times10^9 \times 4\times10^6$

$= \underline{39.2\,W}$

c.) $E_A = P_A \times T_A$

$= 115.2 \times 2.5 \times 10^{-6}$

$= 288\mu J$

$E_B = P_B \times T_B$

$= 39.2 \times 2.5 \times 10^{-6}$

$= 98\mu J$ [2]

d.) Batteries store fixed amount of energy. Therefore, B better, as it uses less energy. [2]

Q3.

a.)

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[4]

b.) It introduces a 1-cycle delay betw. the AND and OR gate. This means that Y is now the OR of the C input on this clock cycle and the AND of inputs A and B, from the previous cycle.

i.e. $Y = C \text{ OR } (A^{-1} \text{ AND } B^{-1})$, where X^{-1} means input X on the previous cycle. [1]

Q4. [10]

a.) Logic synthesis: convert HDL into logical gate netlist (similar to logic diagram). Tool: Xilinx Synthesis Tool (XST)

Partition/Map: divide circuit/netlist up into small partitions that can be implemented on individual LUTs or SLICEs. Tool: Xilinx MAP.

Place and Route: place partitions onto specific SLICEs in the FPGA. Decide wire routing between SLICES. Tool: Xilinx PAR

Create configuration bitstream: convert PAR'd design into a bit-file that will configure all LUTs and switching fabric correctly. Tool: Xilinx BitGen.

Program FPGA: use Impact s/w and a hardware programmer (e.g. JTAG)

b.) Logic synthesis: convert HDL into a logical gate netlist . Tool: Design Compiler
Map to standard cells: map the netlist onto standard cells. Tool: Design Compiler
Place and route: place the standard cells, route the wires between them. Tool: IC Compiler
Create mask: create a physical mask that can be used to etch the ASIC. Tool: done by fab
Fabrication: use the masks to fabricate the ASIC. Tool: done by fab.

Q5

[4]

Let C = cost

$C_{ASIC} = 10^6 + 3x$

$C_{FPGA} = 100x$

Cost crossover at: $10^6 + 3x = 100x$

$97x = 10^6$

$x = 10\,309$

$x < 10\,309$: use FPGAs

$x > 10\,309$: make an ASIC

Q6

[6]

```
module m(
  input A,
  input B,
  input C,
  output Y
  );
  wire w1, w2;
  xor(w1, A, B);
  not(w2, C);
  and(Y, w1, w2);
endmodule
```

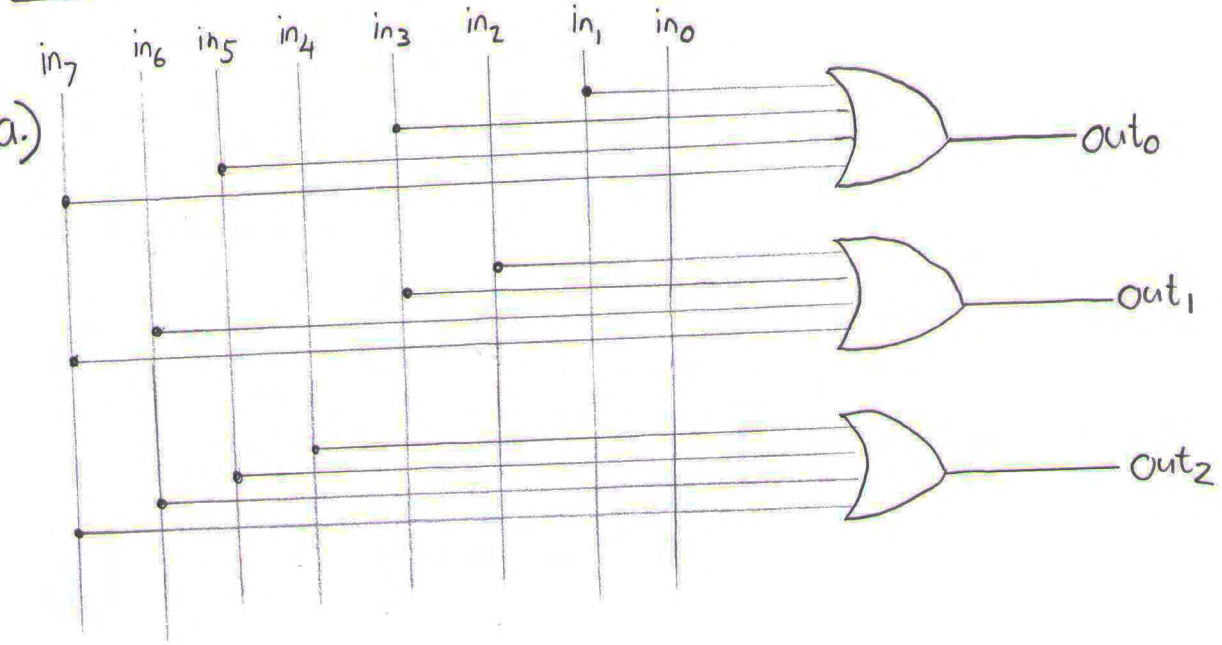
Q7

[5]

```
module m(
  input A,
  input B,
  input C,
  output Y
);
  assign Y = (A ^ B) & (~C);
endmodule
```

Q8.

a.)



[6]

b.)

[4]

```
module oh-to-bin(
  input [7:0] oh-in,
  output [2:0] bin-out
  );
  assign  bin_out[0] = oh_in[7] | oh_in[5] | oh_in[3] | oh_in[1];
  assign  bin_out[1] = oh_in[7] | oh_in[6] | oh_in[3] | oh_in[2] ;
  assign  bin_out[2] = oh_in[7] | oh_in[6] | oh_in[5] | oh_in[4];
endmodule
```