

• Problem 1 •

- If a microprocessor has a 32 bit address bus and a 16 bit data bus, also its registers are 16 bit registers

a) What is the size of the address space this microprocessor can address?

• Because we have a 32 bit address bus we can access  $2^{32}$  different addresses. ∴ The address space is  $2^{32}$  32 bit addresses.

b) This is a 32-bit microprocessor, because the width of the address bus is 32 bits.

• Problem 2 •

End	Start		
1FFF FFFF	0004 0000	= 1FFB FFFF	→ 536608767
2FFF FFFF	2000 8000	= 1FF7 FFFF	→ 33521663
3FFF FFFF	2210 0000	= 1DE FFFF	→ 502267903
40003FFF	4000 2000	= 1FFF	→ 8191
4001FFFF	4001 4000	= 3FFF	→ 4915
40027FFF	4002 6000	= 1FFF	→ 8191
4002BFFF	4002 A000	= 1FFF	→ 8191
4002FFFF	4002 E000	= 7FFF	→ 8191
4003BFFF	4003 A000	= 1FFF	→ 8191
4003FFFF	4003 D000	= 2FFF	→ 12287
4004BFFF	4004 2000	= 9FFF	→ 40959
40057FFF	4005 1000	= 6FFF	→ 28671
400AEFFF	4005 E000	= 50FFF	→ 331775
400F8FFF	400B 0000	= 48FFF	→ 297007
400FBFFF	400F A000	= 1FFF	→ 8191
DF FFFF	4400 0000	= 9B FFFF	→ 2617245695
EC00DFFF	E000 3000	= AFFF	→ 45055
F003FFFF	E000 F000	= 30FFF	→ 200703
FFFFFFFF	E0042000	= 1FFB DFFF	→ 536600575

$$\log_2 (4.227 \cdot 10^9) = 31.977$$

$$\approx 2^{31.977}$$

Addresses are reserved.

Only 67,665,939 are not reserved

4.227301357 · 10<sup>9</sup>  
Addresses are reserved

pg 11

b) - To be backward compatible with future products?  
- Reserved for future use

- To use for constants because it's easier to just grab a memory address than it is to store a constant into memory?

- Allow for compatibility with other products with more peripherals using the same chip.

c)  $0x400A.F000 - 0x400A.F7FF$  or  $0x400A.FFFF$   
START 2KiB END

On memory map this is the end address of EEPROM and key locker but it is 4KiB from the base not 2KiB

d) 16,384 Bits of memory

e) 512 32-bit words

f) 32 BLOCKS

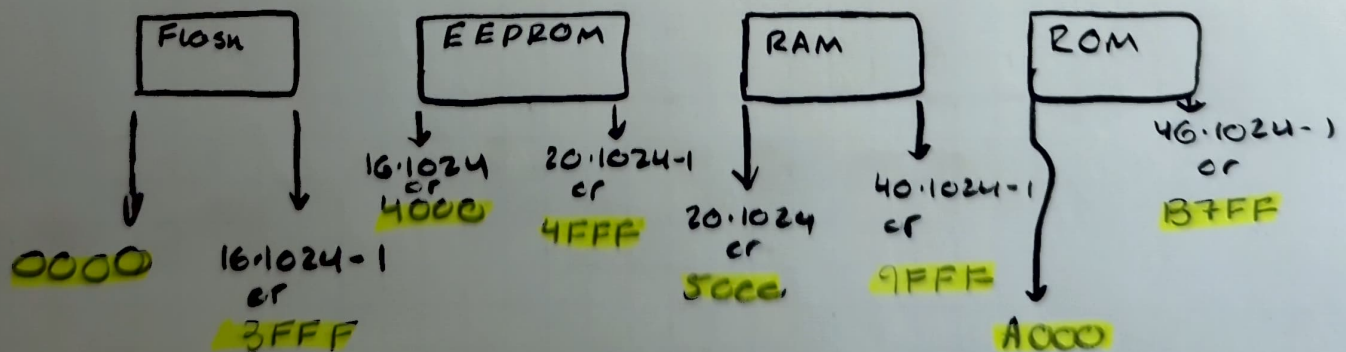
### Problem 30

a) ROM is non volatile

b) The bus interface unit writes data onto the bus during a write cycle.

[Pg 129 5<sup>th</sup> edition]

### Problem 40



• The start and end address of ROM is  $A000$  and  $B7FF$



## Problem 5

LDR R0, Base

LDR R1, [R0, #Offset]

ORR R1, #020

STR R1, [R0, #Offset]

How on earth do we do this in 3 lines?!