School of
Computing Science

CSC1104 – Computer Organisation and Architecture

Tutorials/Laboratory 3:
Internal Memory, External Memory, Raspberry Pi Platform

Assoc. Prof. Cao Qi
Qi.Cao@Glasgow.ac.uk

Objectives

- Exercises of Internal memory and external memory.
- Get familiar with some common Raspberry Pi commands to remote access headless Raspberry Pi platform.

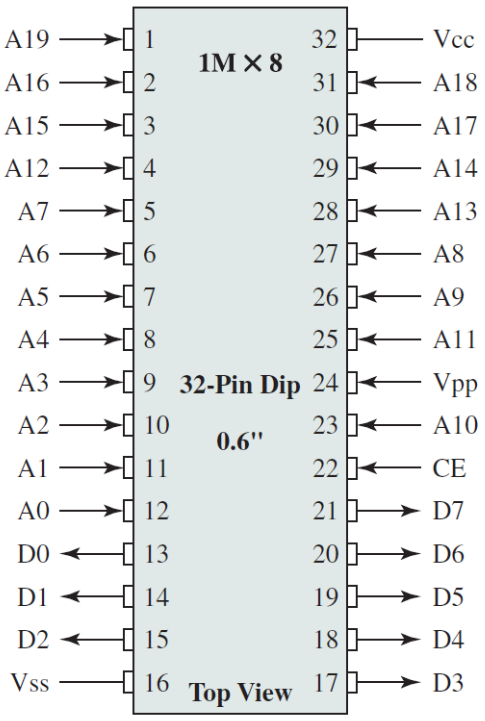
Section 1: Exercise questions for Internal memory and external memory.

1. Different memory organizations can cause different chip packaging, cost, and capacity. Fig. (a) shows an EPROM memory chip (without memory array) with 32 pins. Fig. (b) shows a DRAM memory chip (with memory array) with 24 pins.

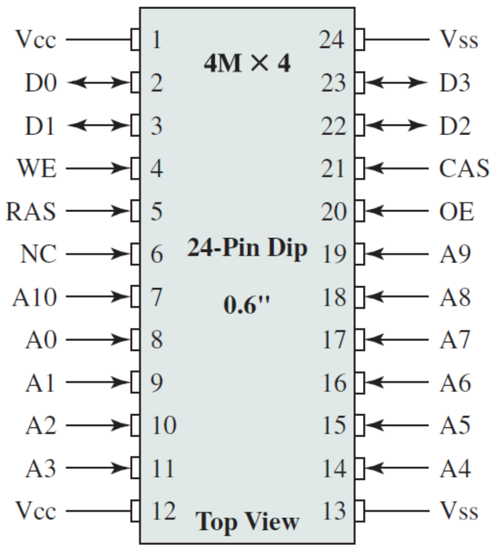
   The EPROM memory organization has 20-bit address bus (A0 – A19) and 8-bit data bus (D0 – D7), as shown in Fig. (*a*).

   The DRAM with memory array organization has 11-bit address bus (A0 – A10) shared by $\overline{RAS}$ and $\overline{CAS}$, as well as 4-bit data bus (D0 –D3), as shown in Fig. (*b*).

   Please prove that the EPROM memory capacity is 1 MiB, and the DRAM memory capacity is 2 MiB.

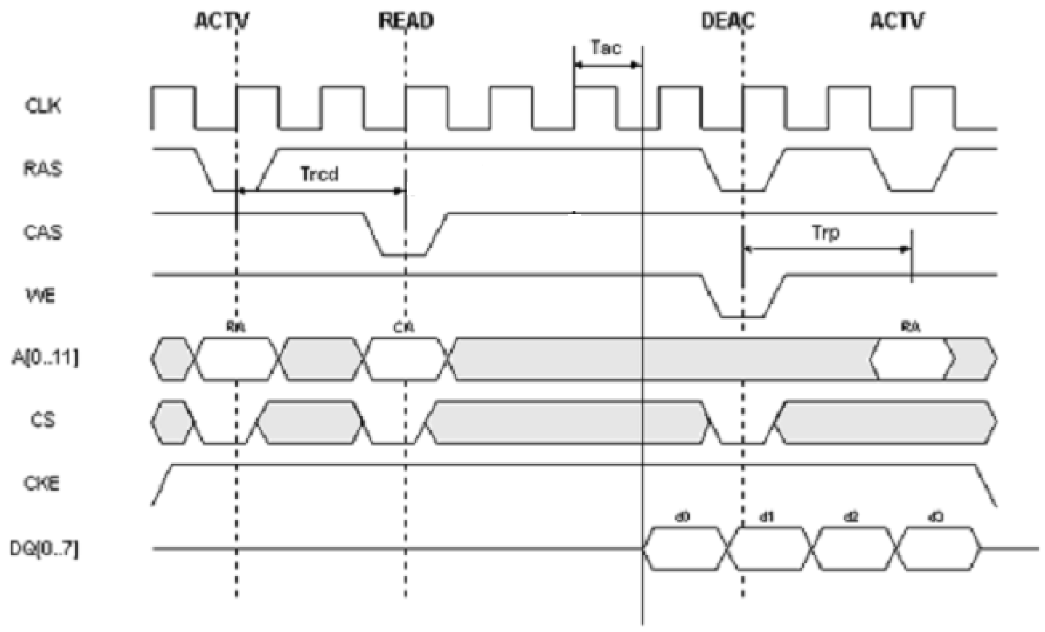


(a) 8-Mbit EPROM



(b) 16-Mbit DRAM

2. For a given DRAM with a burst read operation, the waveform is given below. According to the waveform, please answer these questions:
    a. Is it an asynchronous DRAM or synchronous DRAM, and why?
    b. For this burst ready operation, what is the burst length?
    c. What's the CAS latency for this DRAM?
    d. How big is the memory space capacity (addressable memory size) of this DRAM?



SDRAM burst read timing

Section 2: Continue configuring your Raspberry Pi platform. Please let us know if your Raspberry Pi is not communicating with your laptops.

1. With Putty terminal or ssh terminal access remotely to the Raspberry Pi platform from your laptop, you can now develop your C program, and perform the compilation in Raspberry Pi.

2. **At the terminal of Raspberry Pi**, create a new directory named as project using the command: **mkdir project**

3. **In your laptop**, you can create a **helloworld.c** file, and compile it **using Visual Studio Code in your laptop**.

```
#include <stdio.h>

int main()
{
    printf("\nHello world, my second C code.\n");
    return 0;
}
```

4. Once it is compiled and tested successfully **in your laptop**, you can then transfer the **helloworld.c** file from your laptop to the Raspberry Pi platform using scp commands.

5. For example, the helloworld.c file has been created in the directory of: C:\users\project\helloworld\ in your laptop. **In your laptop**, open a command prompt terminal CMD. Type the commands below to transfer the file.

```
cd C:\Users\user\projects\helloworld
scp helloworld.c pi@xxx.xxx.xxx.xxx:project\helloworld.c
```

   where the xxx.xxx.xxx.xxx denotes the IP address of your Raspberry Pi. The file will be transferred and stored as pi/project/helloworld.c.

6. Switch the Putty ssh terminal for **your raspberry Pi**, please go to the directory of pi/project/. You can check the file **helloworld.c** has been in the folder.

7. At Putty ssh terminal, please use GCC to compile the helloworld.c file using the command below:

```
gcc -o helloworld helloworld.c
```

8. A new file helloworld.exe will be generated. Please execute this file (./helloworld) and check if the results are correct.

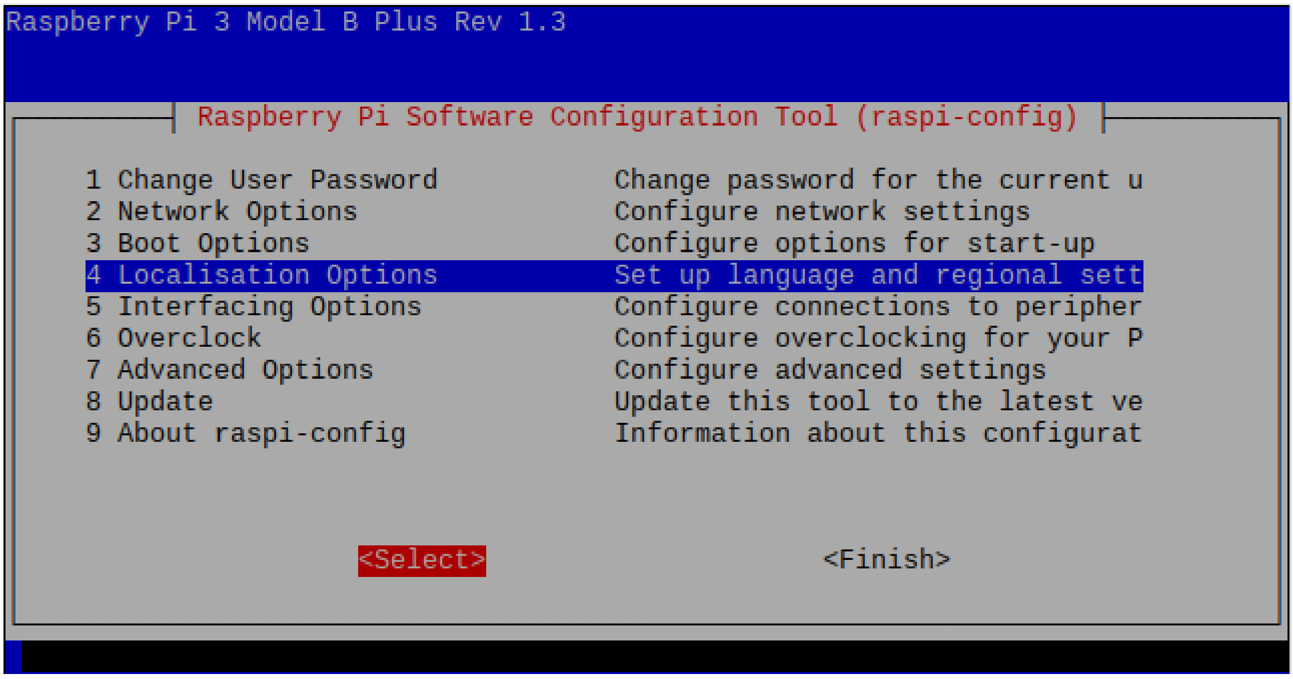
Section 3: Lab manual to remote access and practice several common commands on Raspberry Pi platform.

1. Power on your Raspberry Pi, then connect your Raspberry Pi to your laptop using WiFi connection. Use the method to remote access Raspberry Pi using SSH (such as Putty terminal or Macbook terminal). Type in username and password of Raspberry Pi.

2. An OS is the software used to control and operate a computer. Several common OS include Linux OS, Microsoft's Windows OS, Mac OS of Apple, Android OS from Google, and iOS of Apple, etc).
   The Raspberry Pi platform uses the default operating system (OS), Raspberry Pi OS, that is based on the open-source Linux OS, which recognize Linux commands.

3. Once you SSH and login to the Raspberry Pi remotely from your laptops, type the following commands into the SSH terminal.

   a. **sudo raspi-config** - Please use this command to set the regional location as Singapore.



```
Raspberry Pi 3 Model B Plus Rev 1.3

        Raspberry Pi Software Configuration Tool (raspi-config)

     1 Change User Password          Change password for the current u
     2 Network Options               Configure network settings
     3 Boot Options                  Configure options for start-up
     4 Localisation Options          Set up language and regional sett
     5 Interfacing Options           Configure connections to peripher
     6 Overclock                     Configure overclocking for your P
     7 Advanced Options              Configure advanced settings
     8 Update                        Update this tool to the latest ve
     9 About raspi-config            Information about this configurat

                  <Select>                     <Finish>
```

   b. **ifconfig** - this command is to display the IP address of the Raspberry Pi.

   c. **nano <filename>** - this command is to use the nano text editor to edit a file. After edit, use CTRL + o to save the file. Use CTRL + x to exit.

   d. **sudo nano /etc/dhcpcd.conf** - this command is to view and revise the IP address remotely of Raspberry Pi platform.

   e. **sudo nano /etc/resovl.conf** - this command is to view and revise the IP address of the server.

f. **touch** - this command is to create a new file.

g. **rm** - this command is to delete a file.

h. **cat** - this command is to display the content of a file, e.g. `cat /etc/resovl.conf`.

i. **sudo reboot** - this command is to reboot the Raspberry Pi.

j. **date** - this command is to display date and time of Raspberry Pi platform.

k. If the date of your Raspberry Pi is not correct, please use this command to change the time to the current time: **sudo date -s "Tue Sep 17 2024 10:30"**

l. **pwd** - this command is to display your current working directory.

m. **ls -l** - this command is to display files & contents in the current directory.

n. **cd** - this command is to change to a different directory.

o. **cd ..** - this command is to change to the parent directory of current one.

p. Use the **up arrow key ↑** to view the past used commands one by one.

q. Use **Tab key** when typing command in the terminal mode, such as to select a file in a directory without entering the full name of the file.

r. **mkdir** - this command is to create a new directory.

s. **rmdir** - this command is to delete a directory.

t. **echo** - this command is to line of text/string that are passed as an argument. This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file, e.g. `echo “Welcome to Computer Organization & Architecture”.`

u. **chmod** - this command is to change the permissions for a file. It can use symbols `u` (user that owns the file), `g` (the files group) , and `o` (other users) and the permissions `r` (read), `w` (write), and `x` (execute). Using `chmod u+x *filename*` will add execute permission for the owner of the file. It is useful when we create a script file.

Labs exercise questions in the terminal of Raspberry Pi

4. In Raspberry Pi command line, print out the current working directory:
5. We have created a directory named **project**. Changes the directory into this directory **project**:
6. List the files and sub-folders in current directory:
7. Display the contents of the file helloworld.c in the current directory project:
8. Create a new sub-folder named as **week3** in the current directory:
9. Copy the file named helloworld.c in the current directory project, into the sub-folder **week3**:
10. Change the directory into the sub-folder **week3**:
11. Create a new file named labs3.c in this current directory:
12. List the files and sub-folders in current directory:
13. Delete the file named as helloworld.c in current directory:
14. Edit the file labs3.c in current directory, by adding the C codes into this labs3.c file:

```c
#include <stdio.h>
int main()
{
    printf("\nHello! It is the file created in Raspberry Pi.\n");
    return 0;
}
```

15. Compile this c program labs3.c using gcc compiler, and generate the output file named: **lab3**

16. A new file **labs3** will be generated in the current directory. Execute this file to observe outputs:

17. Return to the parent directory nameds project: