

Accessing IO Registers (1A)

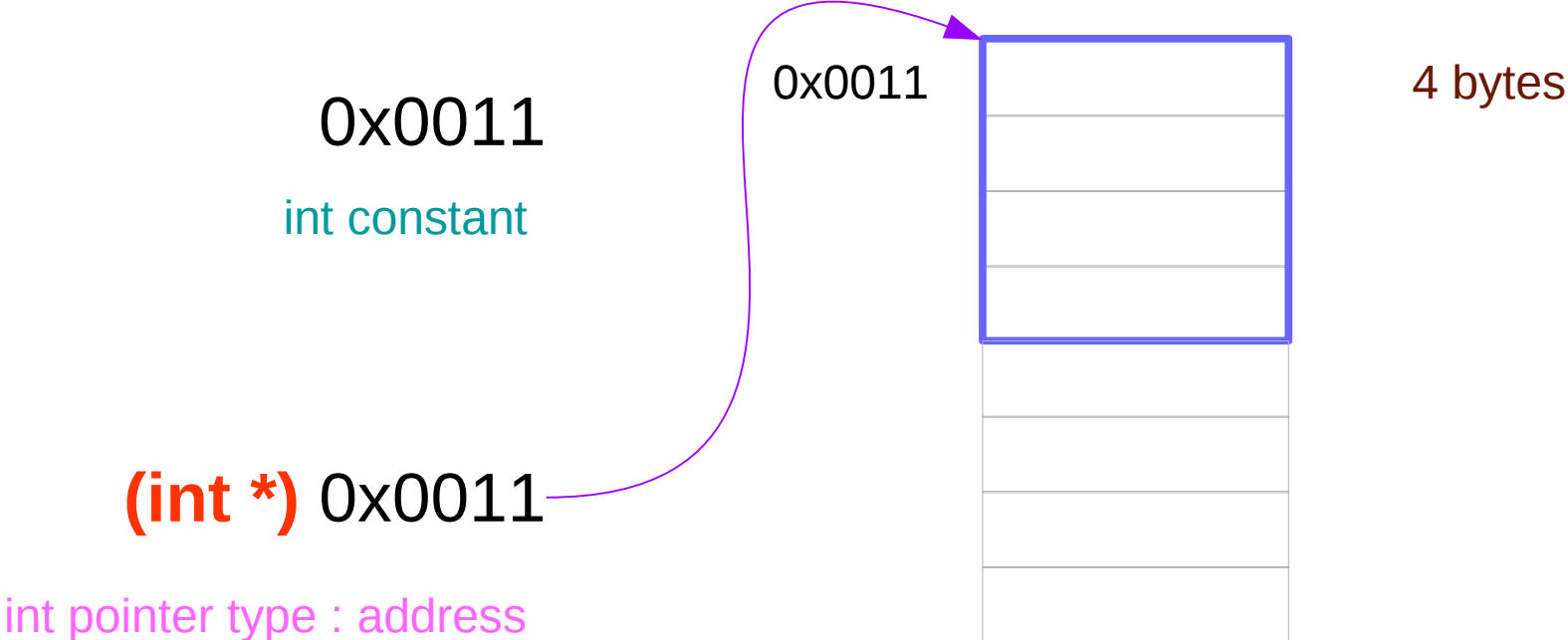
Copyright (c) 2010-2014 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

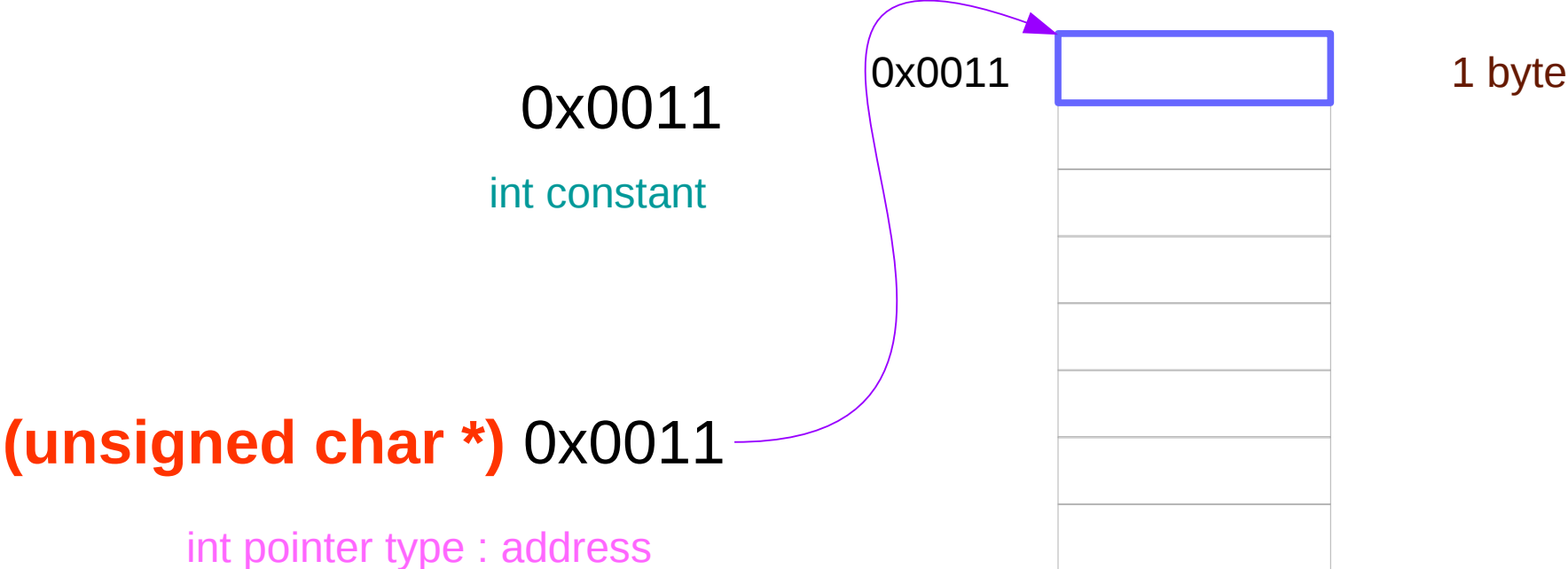
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Address Type Casting (1)

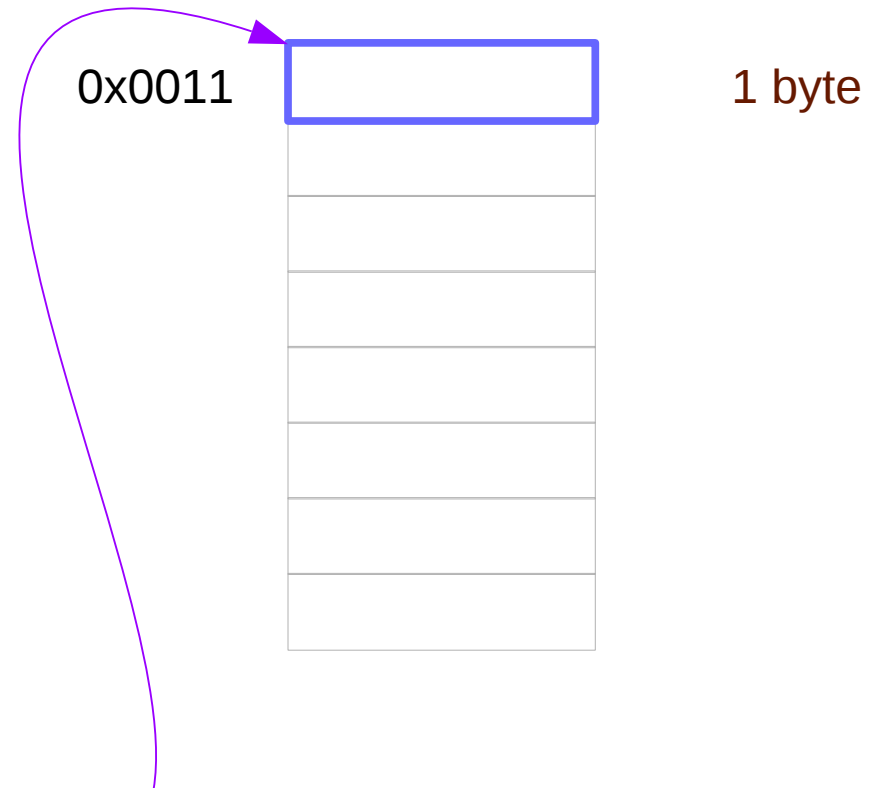


Address Type Casting (2)



Volatile Type Qualifiers

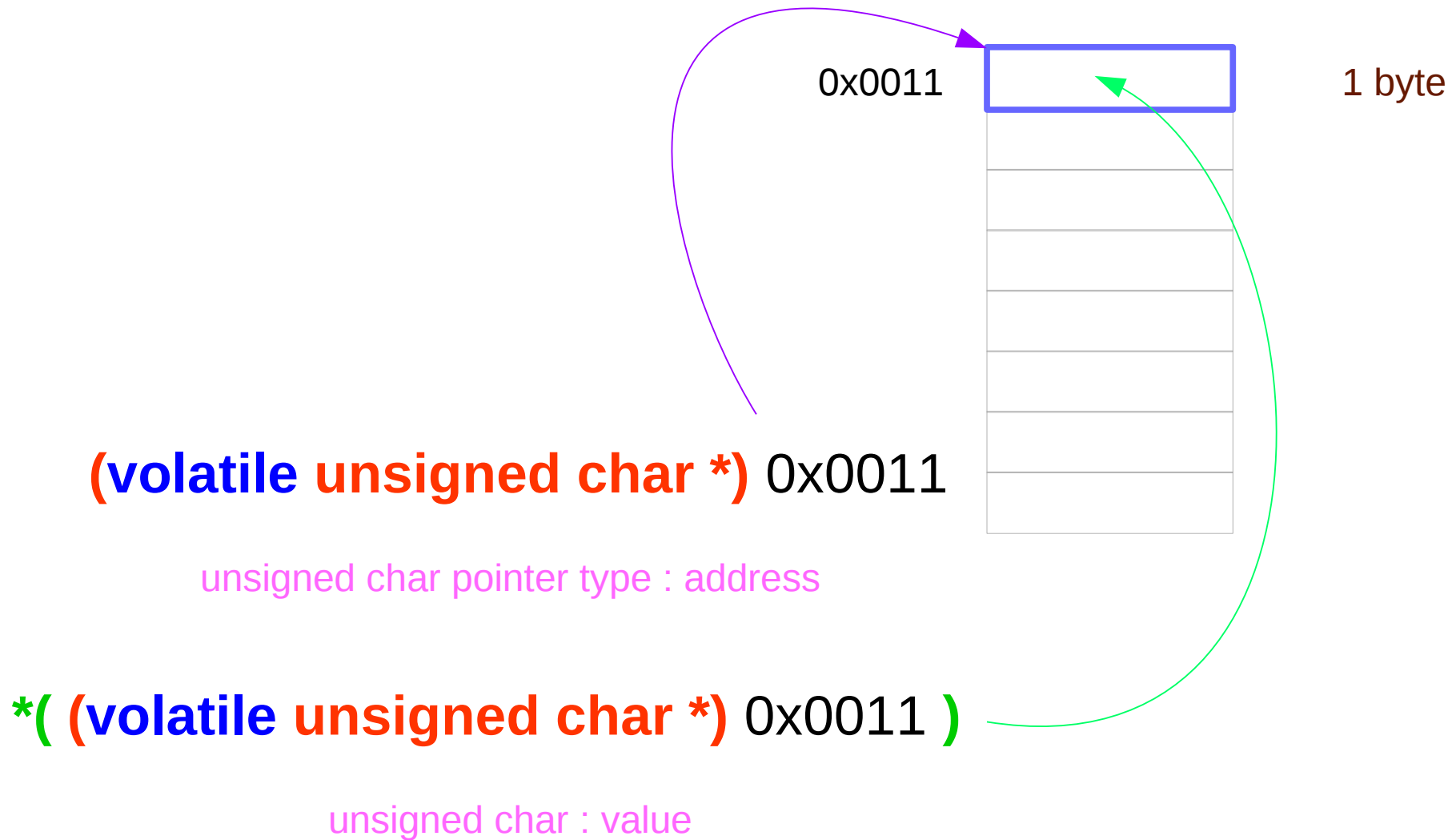
- do not optimize
- provide a reliable access to special memory location used by computer hardware or by asynchronous process such as interrupt handlers



(volatile unsigned char *) 0x0011

unsigned char pointer type : address

Dereferencing Operator *



Macro Definition

```
PORTA = 0x55;
```

```
val = PORTA
```

0x0011

PORTA

1 byte

(volatile unsigned char *) 0x0011

unsigned char pointer type : address

```
#define PORTA *( (volatile unsigned char *) 0x0011 )
```

unsigned char : value

Macro Definition vs. Pointer Variable

```
#define PORTA *(volatile unsigned char *) 0x0011 )
```

unsigned char : value

```
volatile unsigned char * PORTA = (volatile unsigned char *) 0x0011;
```

unsigned char pointer : address

Accessing Multiple Registers

```
typedef unsigned int uint32_t;
```

```
typedef struct {  
    volatile uint32_t reg0;  
    volatile uint32_t reg1;  
    volatile uint32_t reg2;  
    volatile uint32_t reg3;  
} reg_t;
```

```
reg_t * reg_p = (reg_t *) REG_BASE;
```

*pointer variable declaration
with initialization*

```
reg_t * reg_p ;
```

pointer to a structure

REG_BASE

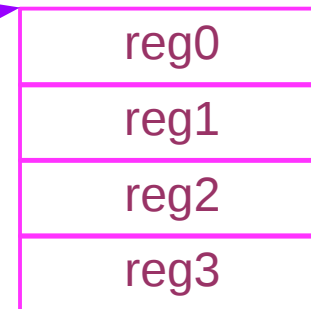


reg_p->reg0

reg_p->reg1

reg_p->reg2

reg_p->reg3



References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun
- [5] “A Whirlwind Tutorial on Creating Really Teensy ELF Executables for Linux”
<http://cseweb.ucsd.edu/~ricko/CSE131/teensyELF.htm>
- [6] “Fundamentals of Embedded Software ...”, D.L. Lewis