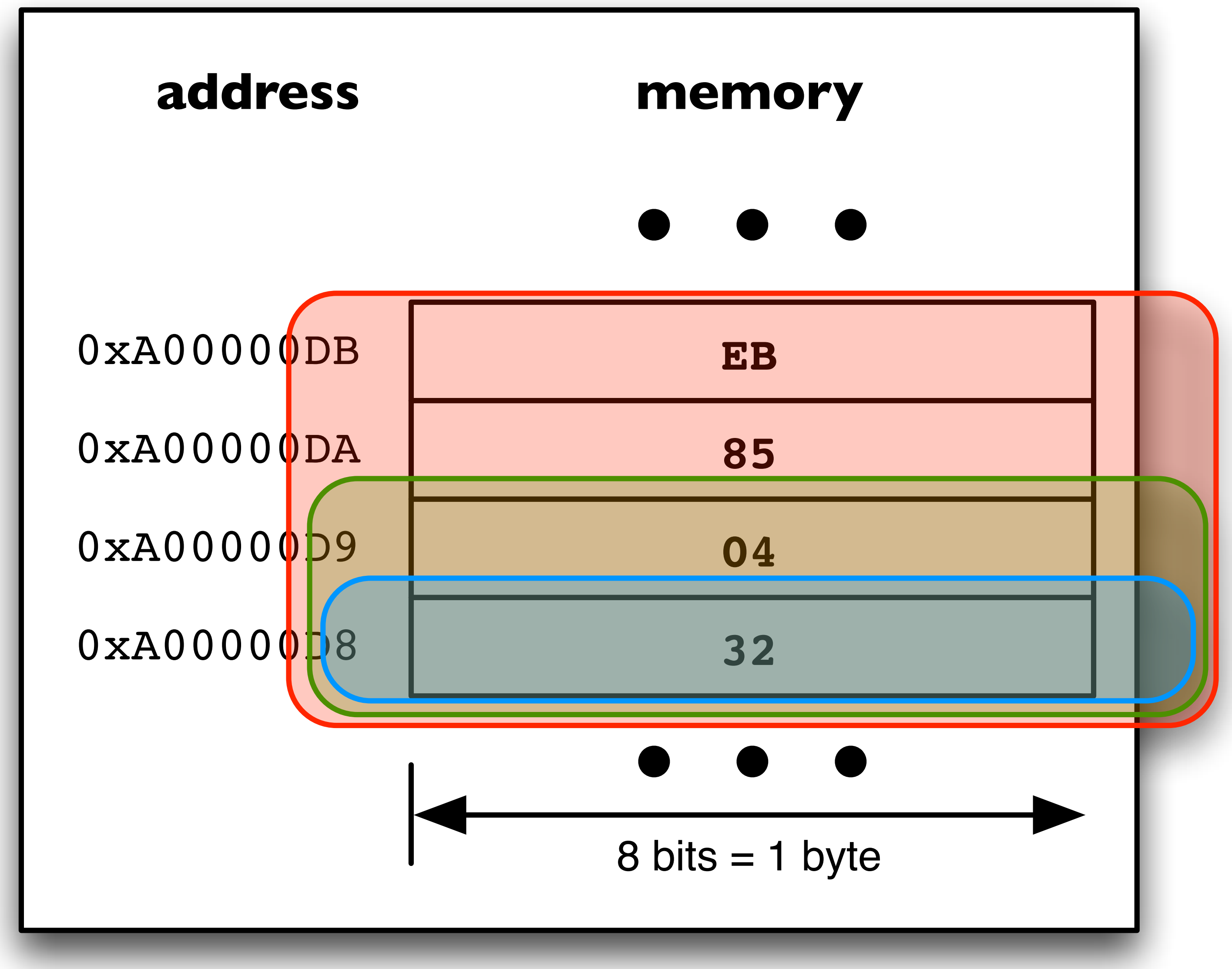
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Memory Re-cap

CS1022 – Introduction to Computing II

Dr Jonathan Dukes / jdukes@scss.tcd.ie
School of Computer Science and Statistics

Load / Store Bytes, Half-words and Words



Byte, **half-word** and **word** at address 0xA00000D8

```
LDR  r0, =0xA00000D8
LDRB r1, [r0]
```

00	00	00	32

```
LDR  r0, =0xA00000D8
LDRH r1, [r0]
```

00	00	04	32

```
LDR  r0, =0xA00000D8
LDR  r1, [r0]
```

EB	85	04	32

Alignment

ARM7TDMI expects all memory accesses to be **aligned**

Examples:

	Address Examples
Word aligned	`0x00000000, 0x00001008, 0xA100000C`
Not word aligned	`0x00000001, 0x00001006, 0xA100000F`
Half-word aligned	`0x00000000, 0x00001002, 0xA100000A`
Not half-word aligned	`0x00000003, 0x00001001, 0xA100000B`

See ARM Architecture Reference Manual Section A2.8

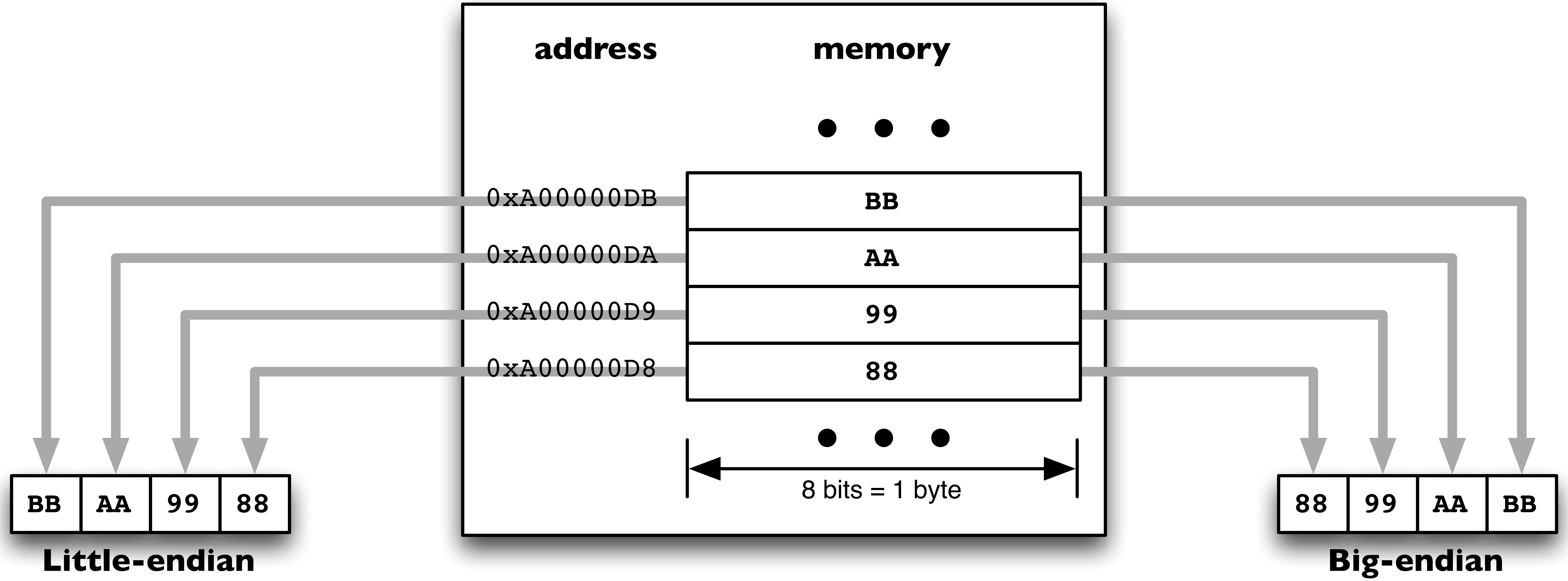
Unaligned accesses are permitted but the result is unlikely to be what was intended

Unaligned accesses are supported by later ARM architecture versions

Endian-ness

Little-endian byte ordering – least-significant byte of word or half-word stored at lower address in memory

Big-endian byte ordering – most-significant byte of word or half-word stored at lower address in memory



Loading Signed Bytes and Half-words

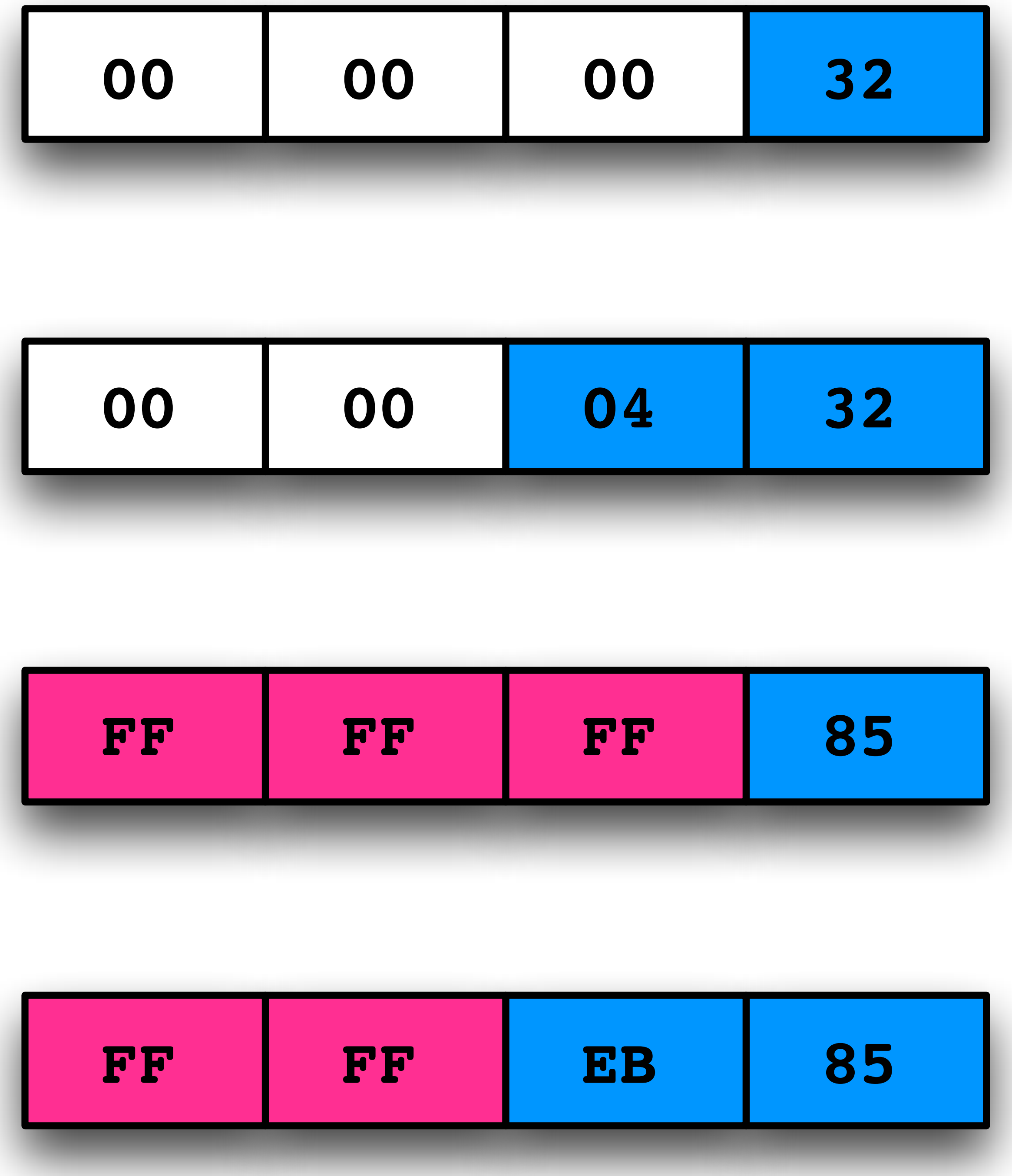
Sign extension performed when loading signed bytes or half-words to facilitate correct subsequent 32-bit signed arithmetic

```
LDR     r0, =0xA00000D8
LDRSB   r1, [r0]
```

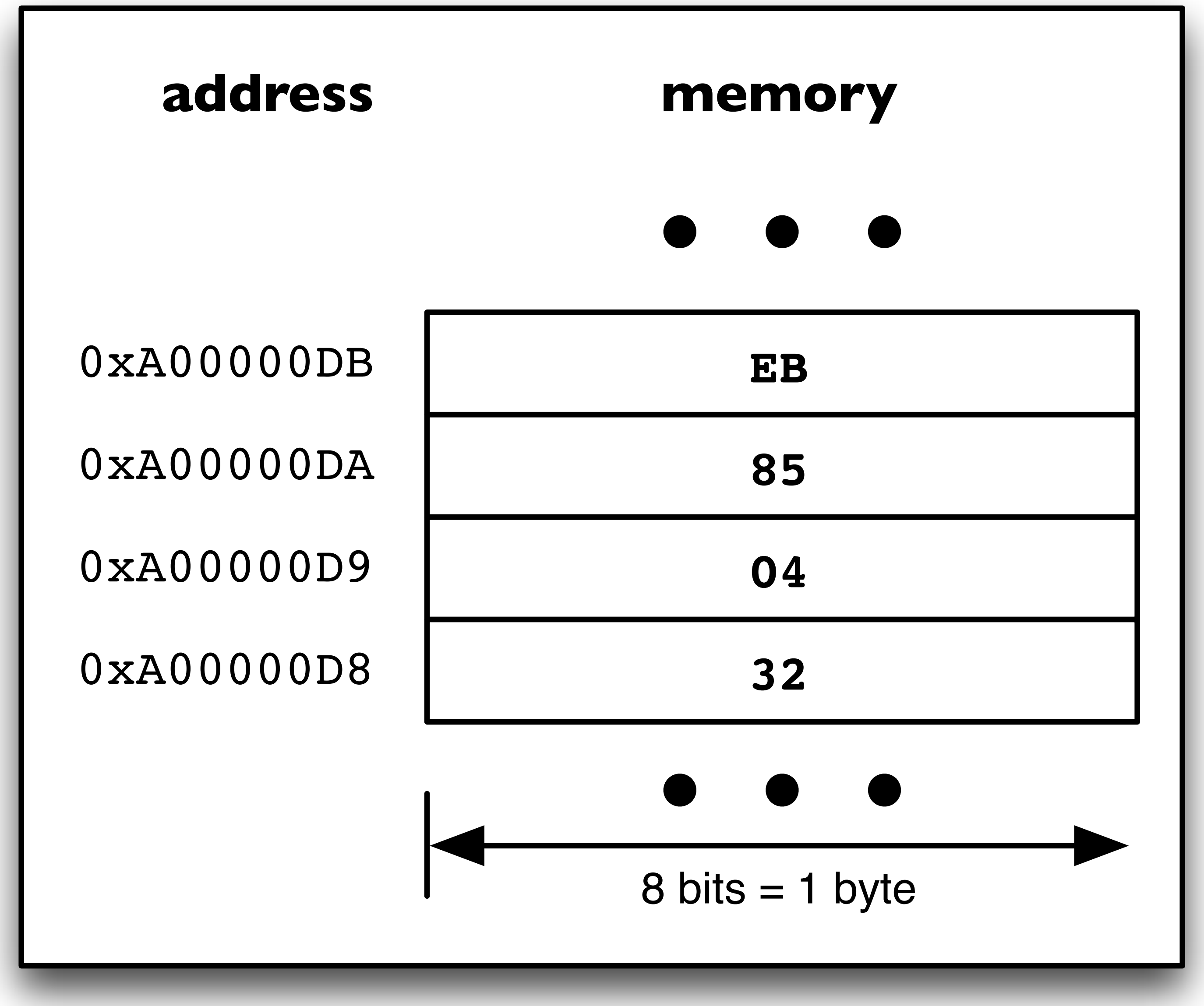
```
LDR    r0, =0xA00000D8
LDRSH r1, [r0]
```

```
LDR     r0, =0xA00000DA
LDRSB   r1, [r0]
```

```
LDR    r0, =0xA00000DA
LDRSH r1, [r0]
```



00	00	00	32
00	00	04	32
FF	FF	FF	85
FF	FF	EB	85



address	memory
	• • •
0xA00000DB	EB
0xA00000DA	85
0xA00000D9	04
0xA00000D8	32
	• • •