

Formato de Instrucción

Gustavo Hurovich

Organización del Computador I - Departamento de Computación
Universidad de Buenos Aires

Conceptos claves del día de hoy

- ▶ Tamaño de memoria
- ▶ Dirección de memoria
- ▶ Unidad direccionable
- ▶ Instrucción
- ▶ Codificación de una instrucción

¿Qué vimos hasta ahora?

- ▶ Cómo representar números Enteros.
 - ▶ Implementamos circuitos lógicos.
 - ▶ Armamos una ALU
 - ▶ Extendimos la ALU para que tenga registros.
-
- ▶ Usamos la máquina de Orga1 (cálculo de desplazamientos + ensamblado, seguimiento).

¿Qué vimos hasta ahora?

- ▶ Cómo representar números Enteros.
- ▶ Implementamos circuitos lógicos.
- ▶ Armamos una ALU
- ▶ Extendimos la ALU para que tenga registros.
- ▶ ???
- ▶ Usamos la máquina de Orga1 (cálculo de desplazamientos + ensamblado, seguimiento).

¿Qué vimos hasta ahora?

- ▶ Cómo representar números Enteros.
- ▶ Implementamos circuitos lógicos.
- ▶ Armamos una ALU
- ▶ Extendimos la ALU para que tenga registros.
- ▶ ??????????????????????????????
- ▶ Usamos la máquina de Orga1 (cálculo de desplazamientos + ensamblado, seguimiento).

¿Qué vimos hasta ahora?

- ▶ Cómo representar números Enteros.
- ▶ Implementamos circuitos lógicos.
- ▶ Armamos una ALU
- ▶ Extendimos la ALU para que tenga registros.
- ▶ ??????????????????????????????
- ▶ Usamos la máquina de Orga1 (cálculo de desplazamientos + ensamblado, seguimiento).

Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

Bienvenido al Mundo Línea. Acá las personas ocupan 1m de ancho, y nada tiene altura. Supongamos una cuadra de unos 32m en este mundo.

Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

Bienvenido al Mundo Línea. Aquí las personas ocupan 1m de ancho, y nada tiene altura. Supongamos una cuadra de unos 32m en este mundo.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

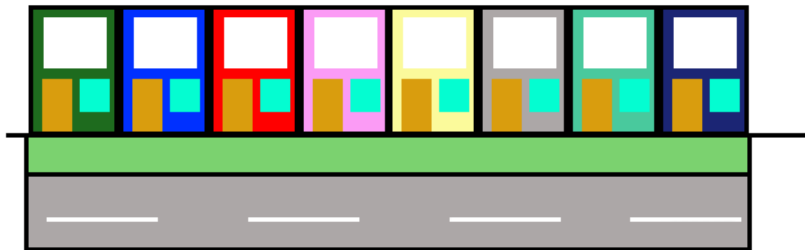
Agreguemos casas en los 32 metros de la cuadra sabiendo que cada casa tiene 4m de ancho.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

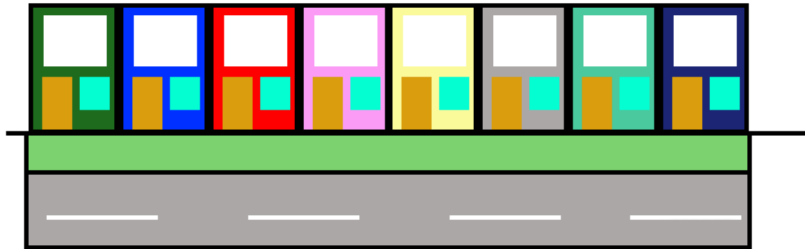
Agreguemos casas en los 32 metros de la cuadra sabiendo que cada casa tiene 4m de ancho.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

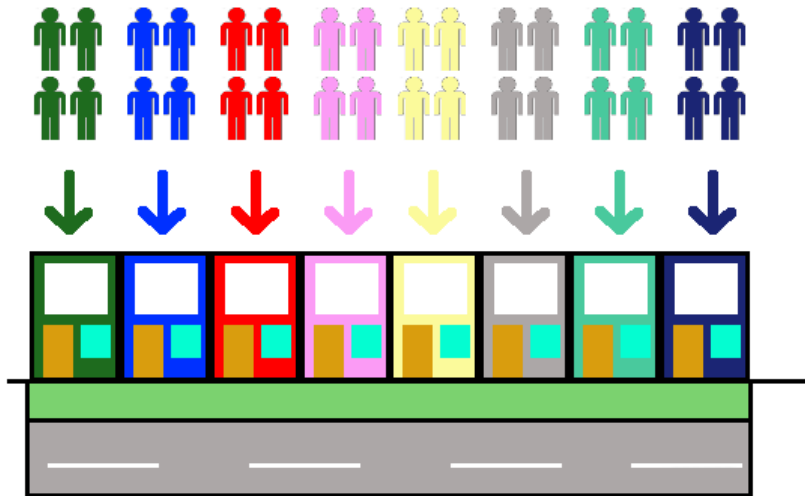
Como una persona ocupa 1m de ancho, y estamos en Mundo Línea donde nada tiene altura, la capacidad de cada casa es de 4 personas. ¿Cuántas personas entran en total en la cuadra?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

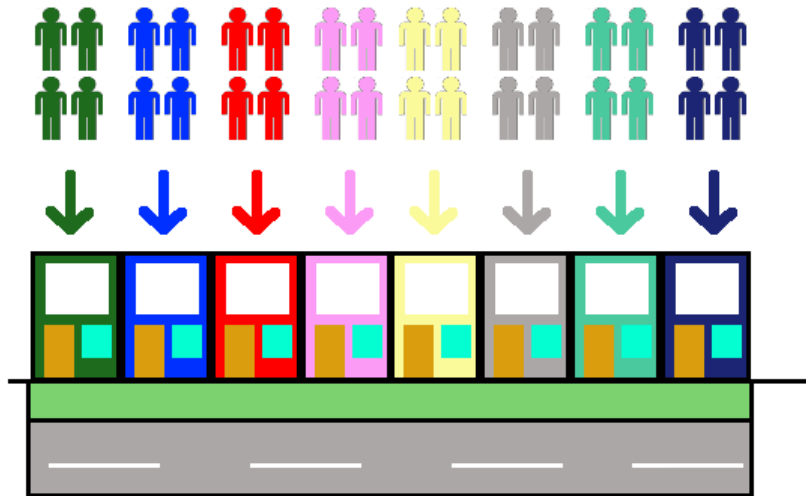
Como una persona ocupa 1m de ancho, y estamos en Mundo Línea donde nada tiene altura, la capacidad de cada casa es de 4 personas. ¿Cuántas personas entran en total en la cuadra?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

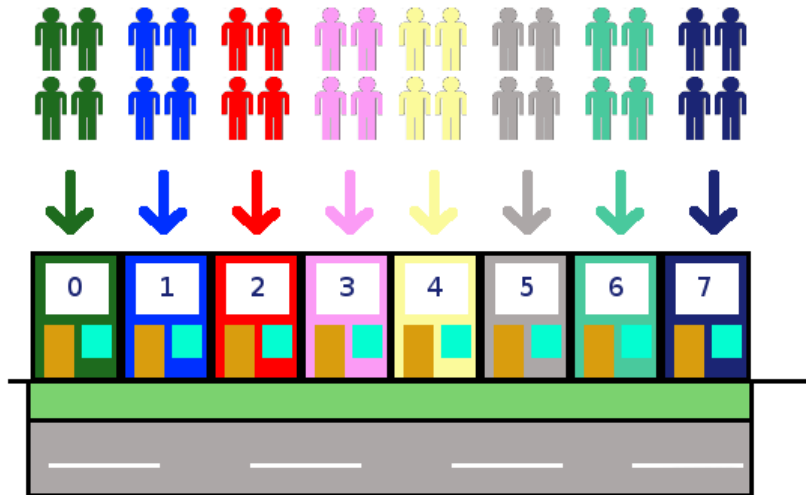
Numeremos las casas.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

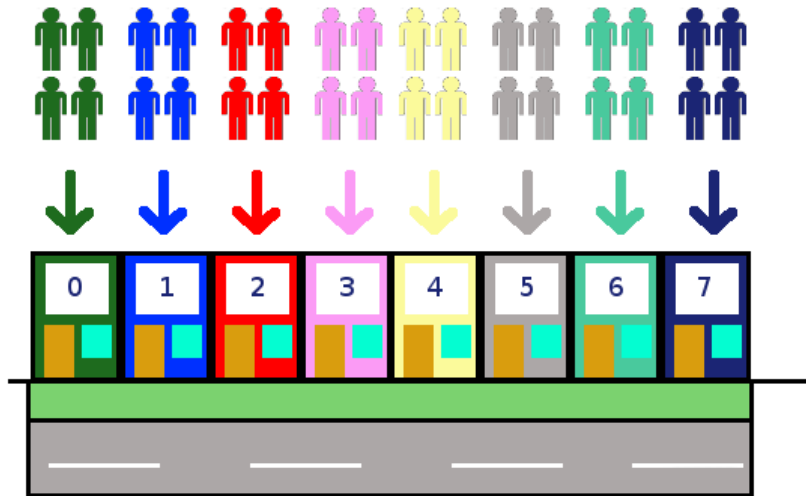
Numeremos las casas.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

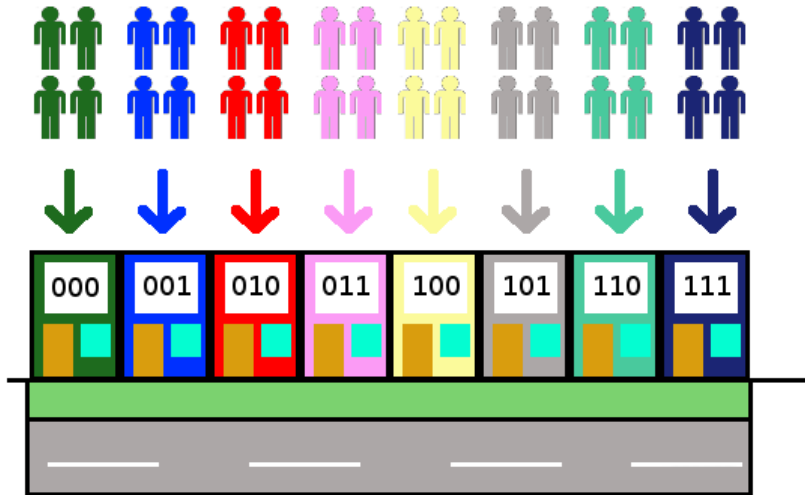
¿Cuántos bits necesitamos para identificar las casas?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

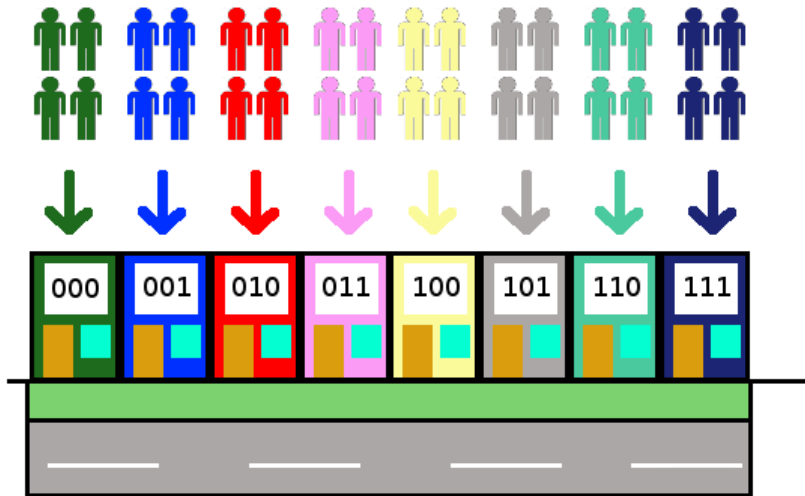
¿Cuántos bits necesitamos para identificar las casas?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

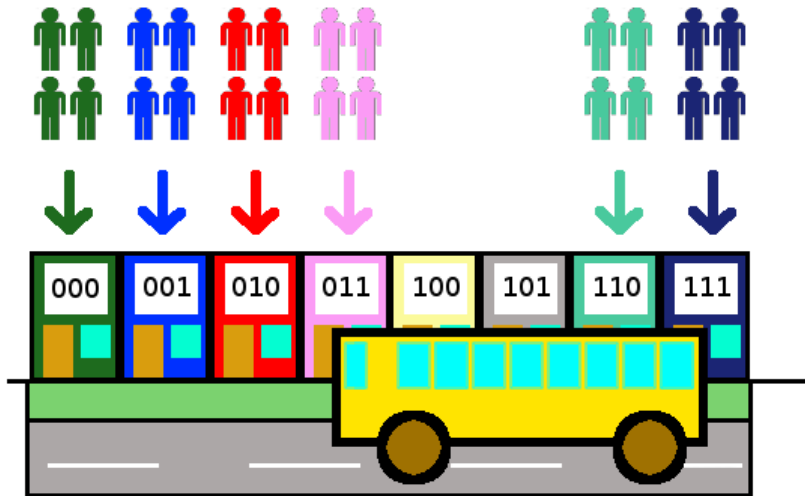
Para ir a trabajar hay un colectivo con lugar para 8 personas que solamente arranca cuando esta lleno. Lleva siempre dos casas contiguas completas.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando?

Para ir a trabajar hay un colectivo con lugar para 8 personas que solamente arranca cuando esta lleno. Lleva siempre dos casas contiguas completas.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

Supongamos que tenemos otra cuadra de 40m.

Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

Supongamos que tenemos otra cuadra de 40m.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

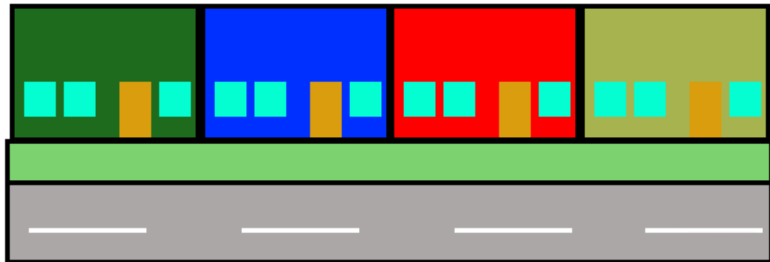
Ahora agreguemos casas a la cuadra de 40m, esta vez de las casas son de 10m de frente.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

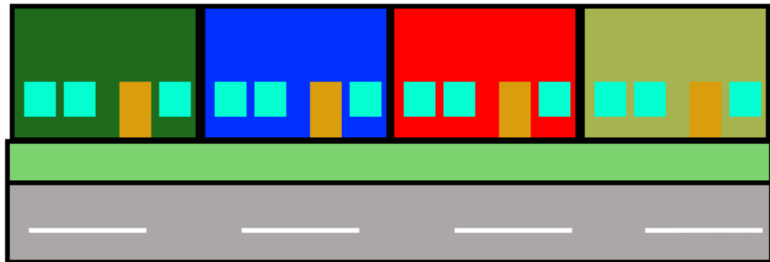
Ahora agreguemos casas a la cuadra de 40m, esta vez de las casas son de 10m de frente.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

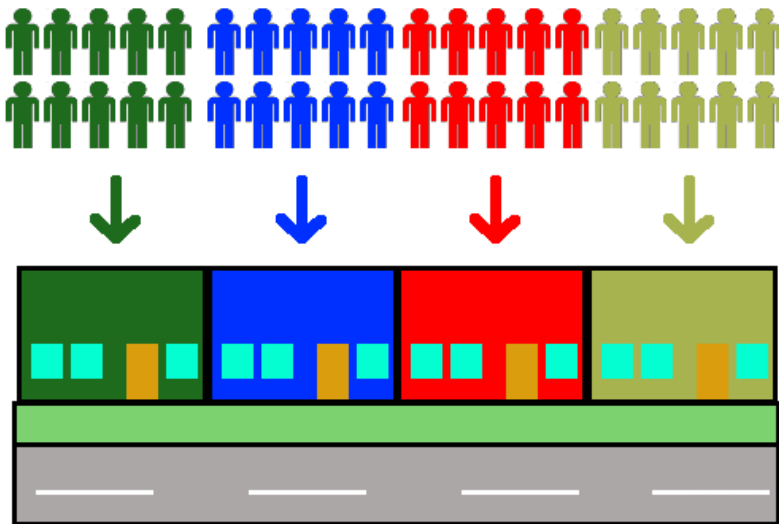
La capacidad de cada casa será de 10 personas. ¿Cuántas personas entran en la cuadra?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

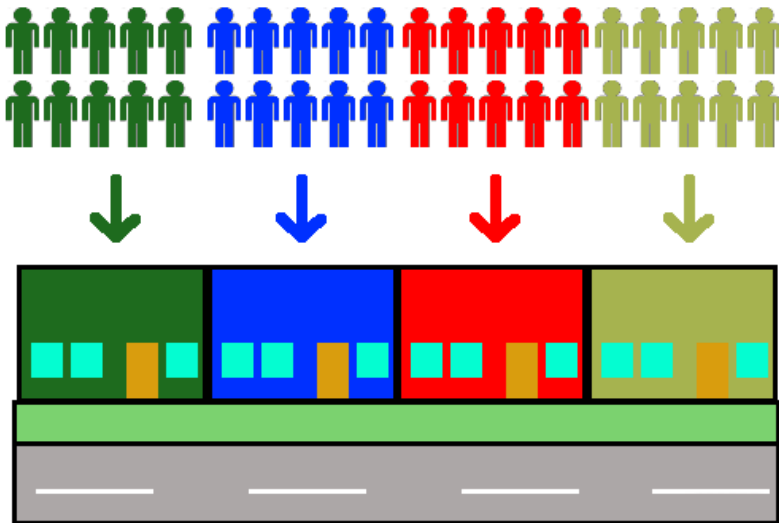
La capacidad de cada casa será de 10 personas. ¿Cuántas personas entran en la cuadra?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

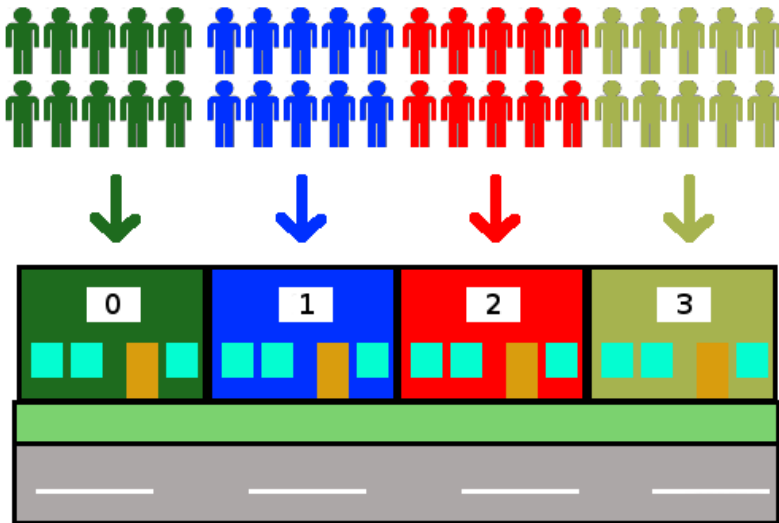
Numeremos las casas.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

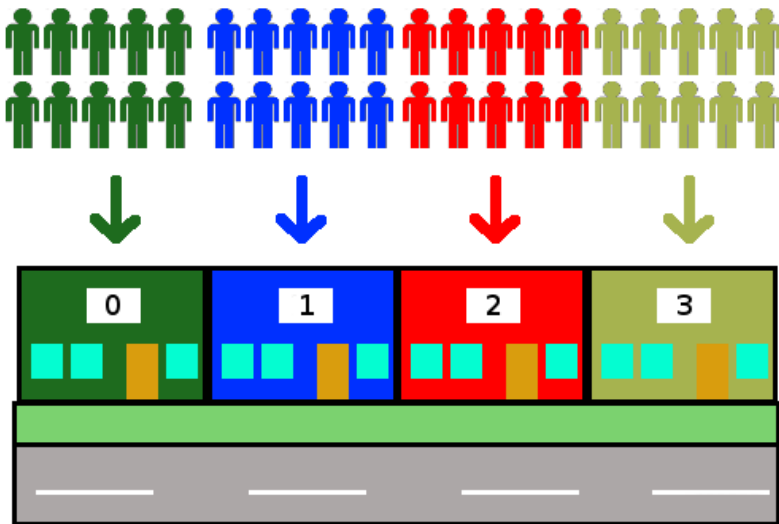
Numeremos las casas.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

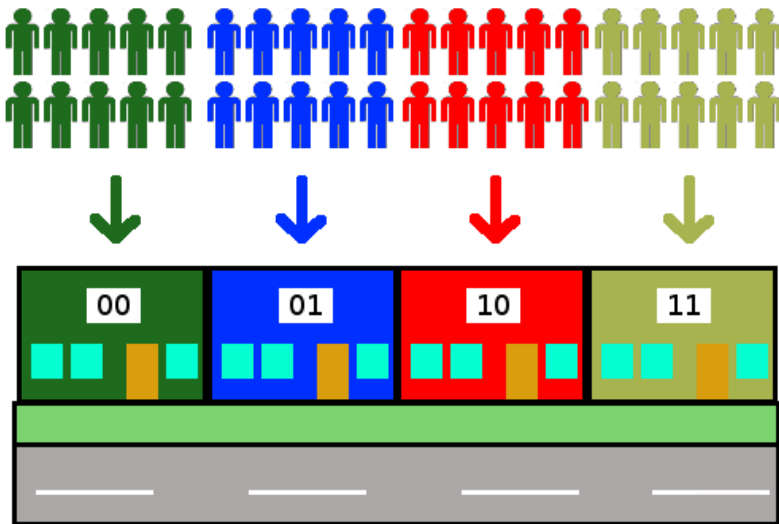
¿Cuántos bits necesitamos para identificar las casas?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

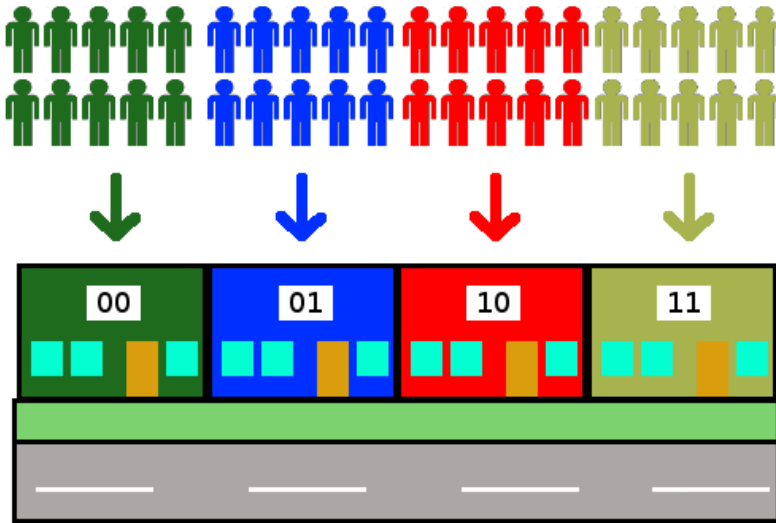
¿Cuántos bits necesitamos para identificar las casas?



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

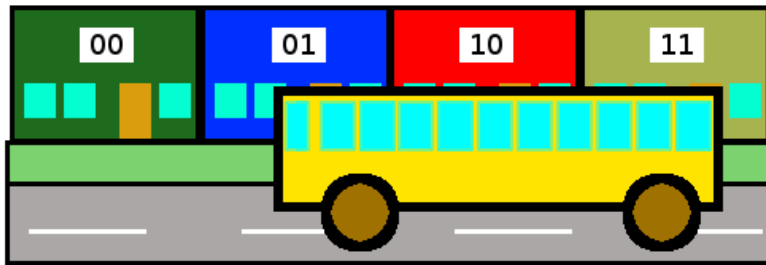
Para ir a trabajar hay un colectivo con lugar para 10 personas que solamente arranca cuando esta lleno. Lleva únicamente a una casa completa por vez.



Tamaño, dirección, unidad direccionable

¿De qué estamos hablando? (toma 2)

Para ir a trabajar hay un colectivo con lugar para 10 personas que solamente arranca cuando esta lleno. Lleva únicamente a una casa completa por vez.



¿Qué tenía esto que ver?

Analogía con la memoria.

- ▶ La cantidad de personas que entra en una cuadra es el tamaño de la memoria.
- ▶ La cantidad de personas que entra en una casa es el tamaño de la unidad direccionable.
- ▶ La cantidad de casas de la cuadra es la cantidad de direcciones de la memoria.
- ▶ La cantidad de bits necesarios para numerar las casas es la cantidad de bits que necesitamos para distinguir las direcciones de memoria.
- ▶ La capacidad del colectivo (o bus) es el tamaño de la palabra.

¡Ahora sí, arranquemos!

¿Cuántos bits son necesarios para poder direccionar la memoria en los siguientes casos?

1. Memoria de 2 GB con direccionamiento a *byte*

“Cuadra de 2 GB, con casas de tamaño 1 *byte*”

- ▶ $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- ▶ $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2. Memoria de 2 GB con direccionamiento a 16 *bits*

- ▶ $\#direcciones = \frac{2 \text{ GB}}{2 \text{ B/dir}} = 2^{30} \text{ dir}$
- ▶ $\#bits \text{ de direccionamiento} = \log_2(2^{30}) = 30 \text{ bits}$

3. Memoria de 16 MB con direccionamiento a *nibble*(4 bits)

- ▶ $\#direcciones = \frac{16 \text{ MB}}{4 \text{ b/dir}} = \frac{2^4 \times 2^{20} \text{ B} \times (8 \text{ b/B})}{4 \text{ b/dir}} = 2^5 \times 2^{20} \text{ dir}$
- ▶ $\#bits \text{ de direccionamiento} = \log_2(2^5 \times 2^{20}) = 25 \text{ bits}$

¿Qué hicimos?

Para saber la cantidad de direcciones que teníamos que direccionar en cada caso, usamos la siguiente fórmula:

$$\text{cantidad de direcciones} = \frac{\text{tamaño de la memoria}}{\text{unidad de direccionamiento}}$$

Una vez encontrada la *cantidad de direcciones de memoria*, utilizamos el logaritmo en base 2 para calcular la *cantidad de bits necesarios* para poder hacer referencia a todas las direcciones de memoria.

Nuestro primer diseño de formato de instrucción

Diseñaremos una nueva máquina de propósito general, llamada TURINGAROUND. La misma cuenta con direcciones de 6 *bits*, palabras de 8 *bits* y direccionamiento a *byte*. Además posee 8 registros de proposito general: $R_0 \dots R_7$ de 8 *bits*.

Nuestro primer diseño de formato de instrucción

El *set* de instrucciones es el siguiente:

Instrucción	Efecto
MOV R_x, R_y	$R_x \leftarrow R_y$
LOAD d	$R_0 \leftarrow [d]$
STORE d	$[d] \leftarrow R_0$
ADD1 R_x	$R_x \leftarrow R_x + 1$
SUB1 R_x	$R_x \leftarrow R_x - 1$
JNE $cte3$	si $R_0 \neq 0$ se incrementa el PC en $cte3$
WAIT	no hace nada
SET	Pone en 0 el valor de todos los registros
ROUND _R	Pone en $R_0 \leftarrow R_7$ y $R_i \leftarrow R_{i-1}$ para $0 < i < 8$
ROUND _L	Pone en $R_7 \leftarrow R_0$ y $R_i \leftarrow R_{i+1}$ para $0 \leq i < 7$

donde $cte3$ es una constante de 3 bits en complemento a 2, y
 $x, y \in [0, 7]$

TURINGAROUND- Memoria

- ▶ ¿Cuál es el tamaño máximo de memoria que podrá tener TURINGAROUND? ¿Cuál debe ser el tamaño (en *bits*) del *program counter* (PC)? ¿Por qué?
- ▶ El tamaño máximo de la memoria será $2^6 \times \text{byte}$, es decir, 64 bytes. El PC tendrá 6 bits puesto que allí se guardan direcciones de memoria y este es el tamaño que se requiere para almacenar una dirección.

TURINGAROUND- Formato de instrucción

- Diseñar un formato de instrucción de longitud fija para el set de instrucciones dado.

Instrucción	Efecto
MOV R_x, R_y	$R_x \leftarrow R_y$
LOAD d	$R_0 \leftarrow [d]$
STORE d	$[d] \leftarrow R_0$
ADD1 R_x	$R_x \leftarrow R_x + 1$
SUB1 R_x	$R_x \leftarrow R_x - 1$
JNE cte3	si $R_0 \neq 0$ se incrementa el PC en cte3
WAIT	no hace nada
SET	Pone en 0 el valor de todos los registros
ROUND R	Pone en $R_0 \leftarrow R_7$ y $R_i \leftarrow R_{i-1}$ para $0 < i < 8$
ROUND L	Pone en $R_7 \leftarrow R_0$ y $R_i \leftarrow R_{i+1}$ para $0 \leq i < 7$

donde cte3 es una constante de 3 bits en complemento a 2, y
 $x, y \in [0, 7]$

► Posible solución:

MOV	= 00	rrr ₁	rrr ₂
LOAD	= 01	mmm	mmm
STORE	= 10	mmm	mmm
ADD1	= 11	000	rrr ₁
SUB1	= 11	001	rrr ₁
JNE	= 11	010	ccc
WAIT	= 11	011	000
SET	= 11	011	001
ROUNDR	= 11	011	010
ROUNDL	= 11	011	011

TURINGAROUND- Extendiendo el formato de instrucción

- ▶ Decidir si es posible agregar más instrucciones sin operandos o no. En caso de ser posible indicar cuántas.
- ▶ Es posible agregar más instrucciones sin operandos. Faltan usar las tiras del 11 011 100 al 11 011 111 (4 en total), y del 11 100 000 al 11 111 111 ($4 \times 8 = 32$). Por lo tanto, se pueden agregar 36 instrucciones sin operandos.

TURINGAROUND- Pseudo-instrucciones

- ▶ Los usuarios de TURINGAROUND van a necesitar muy frecuentemente sumarle a un registro el valor de una constante, pero el set de instrucciones no provee la operación “ADD R_x , N ” cuyo efecto es $R_x \leftarrow R_x + N$, que se usará para N igual a 2, 3 ó 7. Dado que NO es posible modificar el set de instrucciones, ¿es posible dotar de un mecanismo para que el programador pueda hacer uso de esta funcionalidad? ¿Cómo?
- ▶ Es posible. Simplemente hay que repetir la instrucción ADD1 R_x 2, 3 ó 7 veces en función de cuánto valga N . Entonces, en el código del ensamblador lo que vamos a tener es que primero chequea el valor de N y luego decide cuántas veces debe ejecutar ADD1 para poder simular el ADD R_x N .

Preguntas

¿Preguntas?

Lo que viene

Con lo visto hasta ahora pueden realizar la práctica 4.

Martes 3 de octubre 17hs. Taller de Seguimiento.