Nivel ISA – Parte 4

Arquitectura y Organización de Computadoras II

Ticiano J. Torres Peralta

REV2021

Las 7 Dimensiones del ISA

Que tiene que considerar un diseñador de ISA?

- La clase del ISA
- El modelo de memoria
- Los modos de direccionamiento
- Tipos y tamaños de datos (y de operandos)
- Tipos de operaciones
- Tipos de control de flujo
- **Formato de instrucción**
- ***Registros Disponibles* (Arquitecturas GPR)**

Formato de Instrucción

Instrucciones de un ISA están formados por un código, llamado el opcode o código de operación, y información adicional, llamada operando, que indica adonde se encuentra el dato que va a ser manipulado.

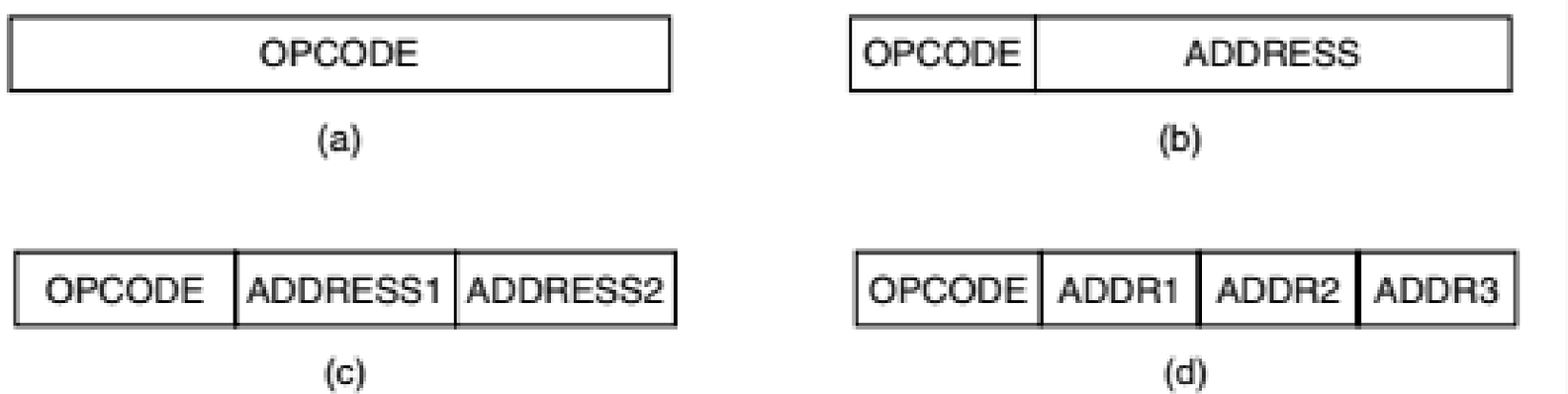


Figure 5-9. Four common instruction formats: (a) Zero-address instruction. (b) One-address instruction (c) Two-address instruction. (d) Three-address instruction.

Formato de Instrucción

El formato de instrucción es de suma importancia en la definición del ISA. El mismo indica de que forma la instrucción va a ser codificada en una representación binaria que entienda y ejecute el procesador.

Esta representación tienen que ser decodificada por el procesador para rápidamente encontrar la operación y sus operandos.

Formato de Instrucción

Una importante decisión es como codificamos los modos de direccionamiento. Esto depende de el rango de modos de direccionamiento, y el grado de independencia entre el mismo y el opcode.

Computadoras antiguas tenían lo que se llama especificador de dirección (address specifier), extra bits que acompañan cada operando que indican cual es el modo de direccionamiento del mismo. Las ISAs donde cada operación soportaba cada modo de direccionamiento, se consideraban ortogonales.

Hoy, en arquitecturas load-store, un máximo de un operando puede direccionar memoria con solo uno o dos modos de direccionamiento. En este caso, es conveniente codificar el modo de direccionamiento como parte del opcode.

Formato de Instrucción

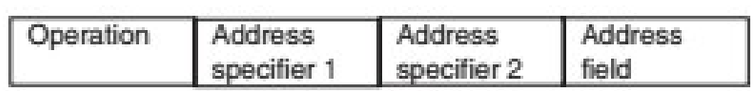
Otro decisión que hay que tomar es si la longitud va a ser fija o variable.

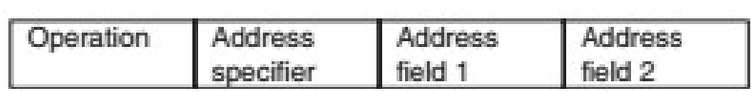
- Variable: un estilo mas viejo pero flexible, util cuando hay que definir muchos modos de direccionamiento y operandos. Produce programas mas compactos.
- Fija: Combina la operación y el modo de direccionamiento en el opcode. Es muy común que definan una sola longitud para todas las instrucciones del ISA. Mas útil cuando hay pocos modos de direccionamiento y operaciones. Es mas fácil de decodificar.

```
add EAX,1000(EBX)
```

(a) Variable (e.g., Intel 80x86, VAX)

(b) Fixed (e.g., Alpha, ARM, MIPS, PowerPC, SPARC, SuperH)





(c) Hybrid (e.g., IBM 360/370, MIPS16, Thumb, TI TMS320C54x)

Formato de Instrucción

Consideraciones para la longitud de la instrucción:

- Por lo general, mientras mas corta la instrucción, mejor. La memoria hoy en día es significantemente mas lenta que el procesador, si el procesador puede buscar múltiples instrucciones en un solo pedido, mejor. Crear una limitación en la cantidad que se puede direccionar.
- Suficientes bits en el opcode para expresar todas las operaciones deseadas. (y permitiendo una cantidad no implementada para futura expansión).
- La cantidad de bits que definen el operando teniendo en cuenta la cantidad de registros y el modelo de memoria.

Formato de Instrucción

El diseñador tienen que balancear:

- El impacto del tamaño del campo de registro y modo de direccionamiento en el tamaño de la instrucción y por consecuencia el tamaño del programa.
- Tener instrucciones con una longitud apropiada para evitar limitaciones del hardware y tomar provecho de técnicas para mejorar rendimiento. Por lo general, esto significa una longitud en multiples de bytes o palabras.

Formato de Instrucción

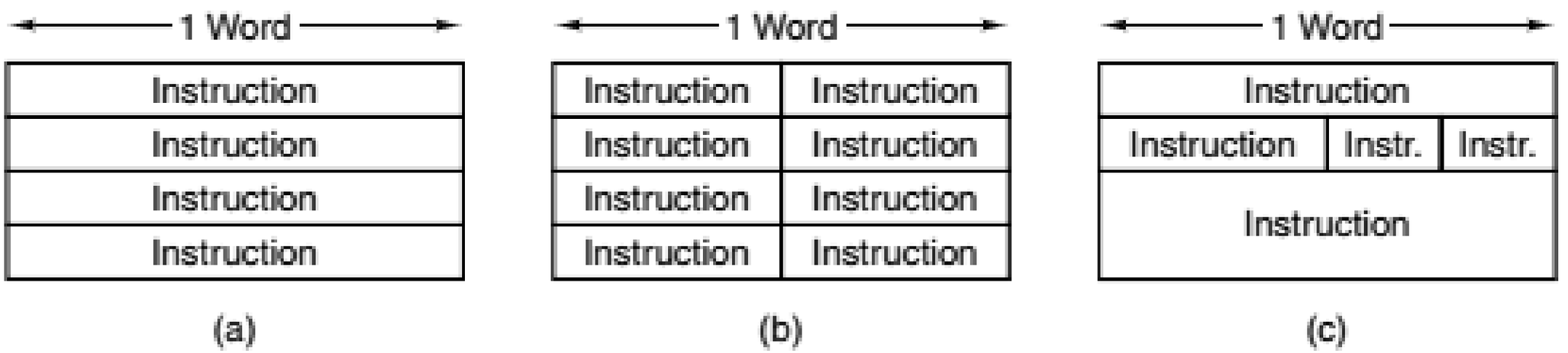


Figure 5-10. Some possible relationships between instruction and word length.

Formato de Instrucción

Considerando una instrucción de longitud fija, siempre hay una pelea entre la cantidad de instrucciones disponibles y la cantidad de direcciones que podemos direccionar.

Si tenemos una instrucción de k + n bit, con:

- k bits para opcode.
- n bits para direccionar un operando.

Hay 2^k posibles operaciones que pueden diseccionar hasta 2^n celdas de memoria/registros.

Alternativamente podemos tener una instrucción de k + n bits, con:

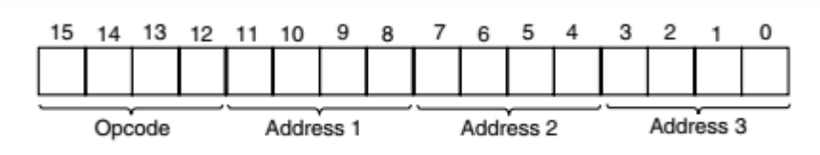
- k – 1 bits para opcode.
- n + 1 bits para direccionar un operando.

Ahora tenemos la mitad de posibles operaciones, pero el doble para direccionar el operando.

Entendiendo esto nos da lugar a entender una técnica que nos permite aprovechar la cantidad limitada de bits que tenemos para definir cada campo. La misma se llama Expansión del Código de Operacion.

Formato de Instrucción

Supongamos que tenemos que diseñar instrucciones de 16 bits

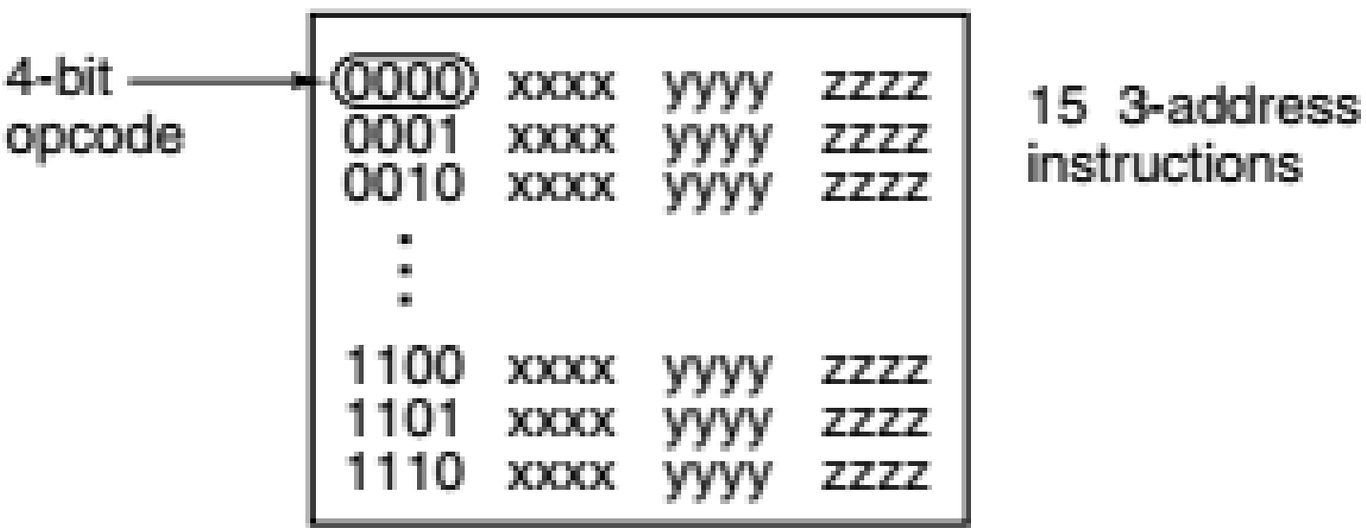


Y queremos:

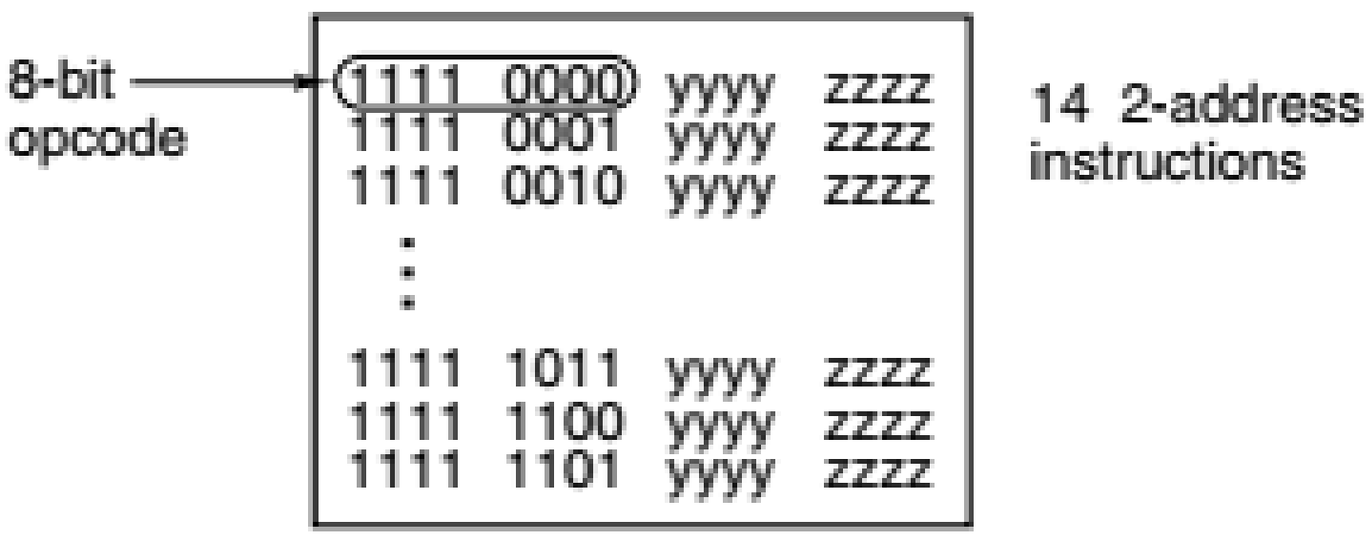
- 15 instrucciones de 3 direcciones
- 14 instrucciones de 2 direcciones
- 31 instrucciones de 1 direcciones
- 16 instrucciones de 0 direcciones.

Un total de 76 instrucciones.

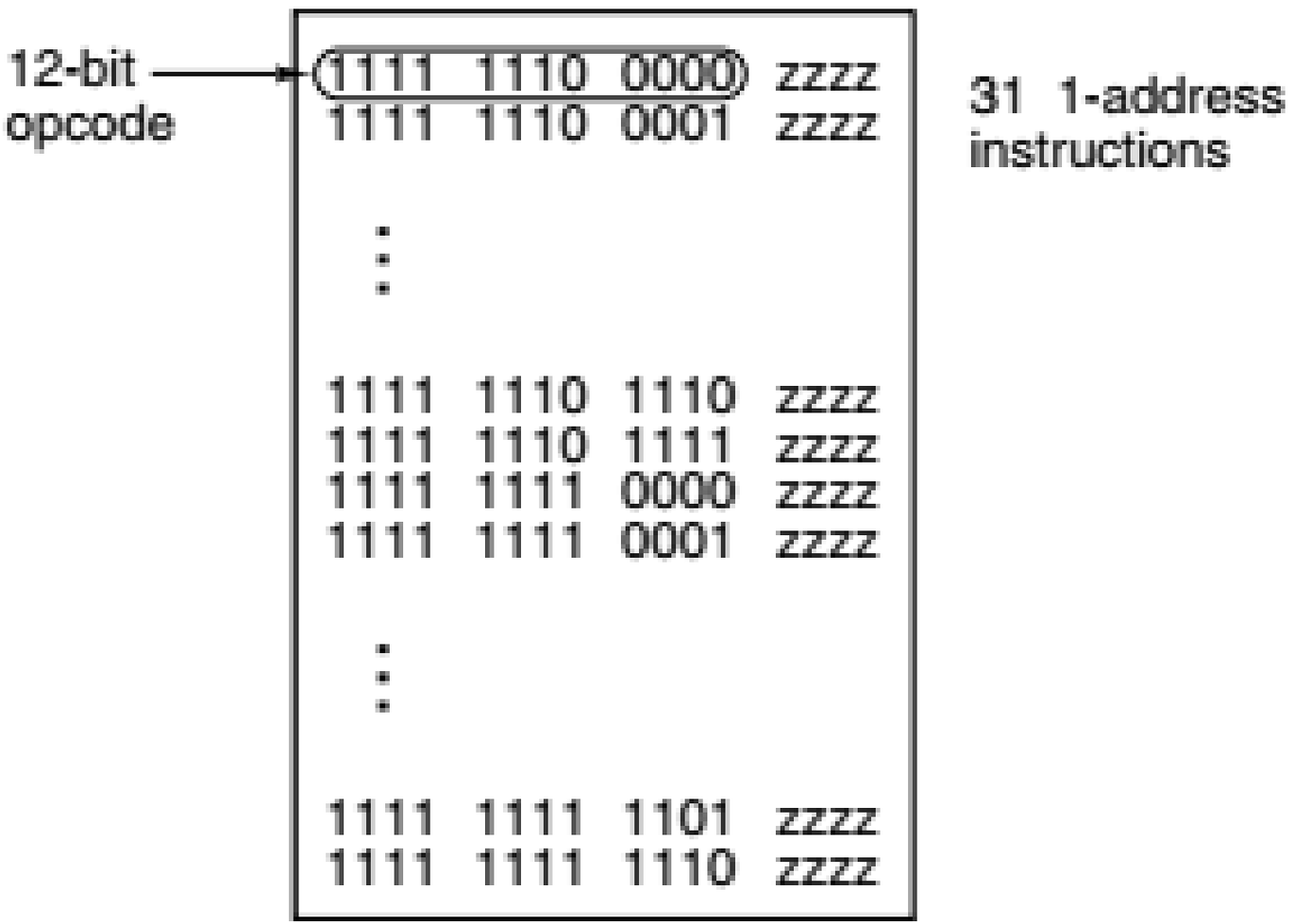
Formato de Instrucción



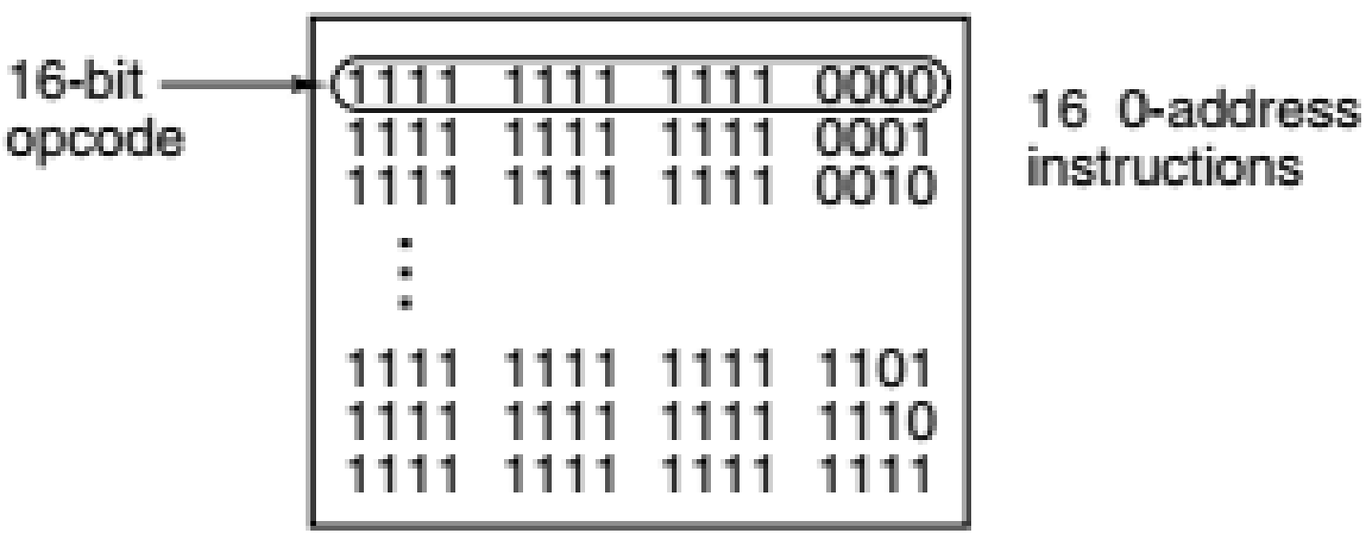
Formato de Instrucción



Formato de Instrucción



Formato de Instrucción



VLIW (Un ejemplo moderno)

- La familia de procesadores Crusoe usan un formato de instrucción especial, llamado VLIW (Very Long Instrucción Word).
- En VLIW, el compilador o un pre-processador organiza las operaciones de tal manera que puedan ser operadas en paralelo y compone una instrucción muy larga.
- Con esta instrucción, el procesador ya no tienen la necesidad de analizar sus partes y puede despachar y ejecutarlas en paralelo.

VLIW (Un ejemplo moderno)

- La ventaja principal de VLIW, es que parte de la complejidad de descubrimiento y planificación es movida al compilador.
- Esto significa que la arquitectura puede ser mas simple, mas barata y que requiera menos potencia.
- Crusoe:
  - Usa VLIW ensambladas por un pre-procesador en su memoria flash.
  - No necesita la habilidad de descubrir y planificar operaciones en paralelo.
  - Contienen un cuarto de los transistores de un procesador normal.
  - Puede ser operador por una pila por casi un día entero.
  - Emula el ISA x86.

Formato de Instrucción

Los factores a considerar son varios y llevan un peso grande en la decisión, ya que un ISA exitoso tiene que poder perdurar por muchos años.

Al final del día, se tiene que ofrecer:

- La posibilidad de expandir el ISA con nuevas instrucciones.
- Poder implementar el ISA de forma eficaz con avances en tecnología.
- Que el costo de implementación no supere lo esperado por mercado.