

# Arquitectura de Computadores

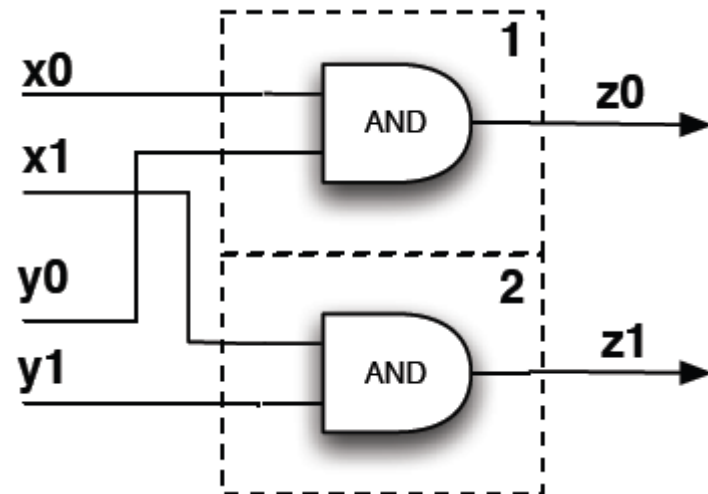
## CC4301

### Clase 8: Multiplicadores

Semestre Primavera 2013  
Profesor: Pablo Guerrero

# Elementos con Múltiples Bits

- Generalizaciones a varios bits de elementos de un bit.
- Implementación: Paralelo.
- Ej:
  - Compuertas
  - Cod-Decod
  - Mux-Demux



# Observación

- Notar que en cualquier base (N):
  - $x \ll m = xN^m$ 
    - En decimal:  $2 \ll 3 = 2000 = 2 \times 10^3$
    - En binario:  $11 \ll 2 = 1100 = 11_2 \times 2^2$
- Por lo tanto:
  - $[x_k \dots x_0] = \sum_{i=0}^k x_i N^i$ 
    - En decimal:  $502 = 5 \times 10^2 + 0 \times 10^1 + 2 \times 10^0$
    - En binario:  $1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

# Algoritmo “de Colegio”

Para Decimales:

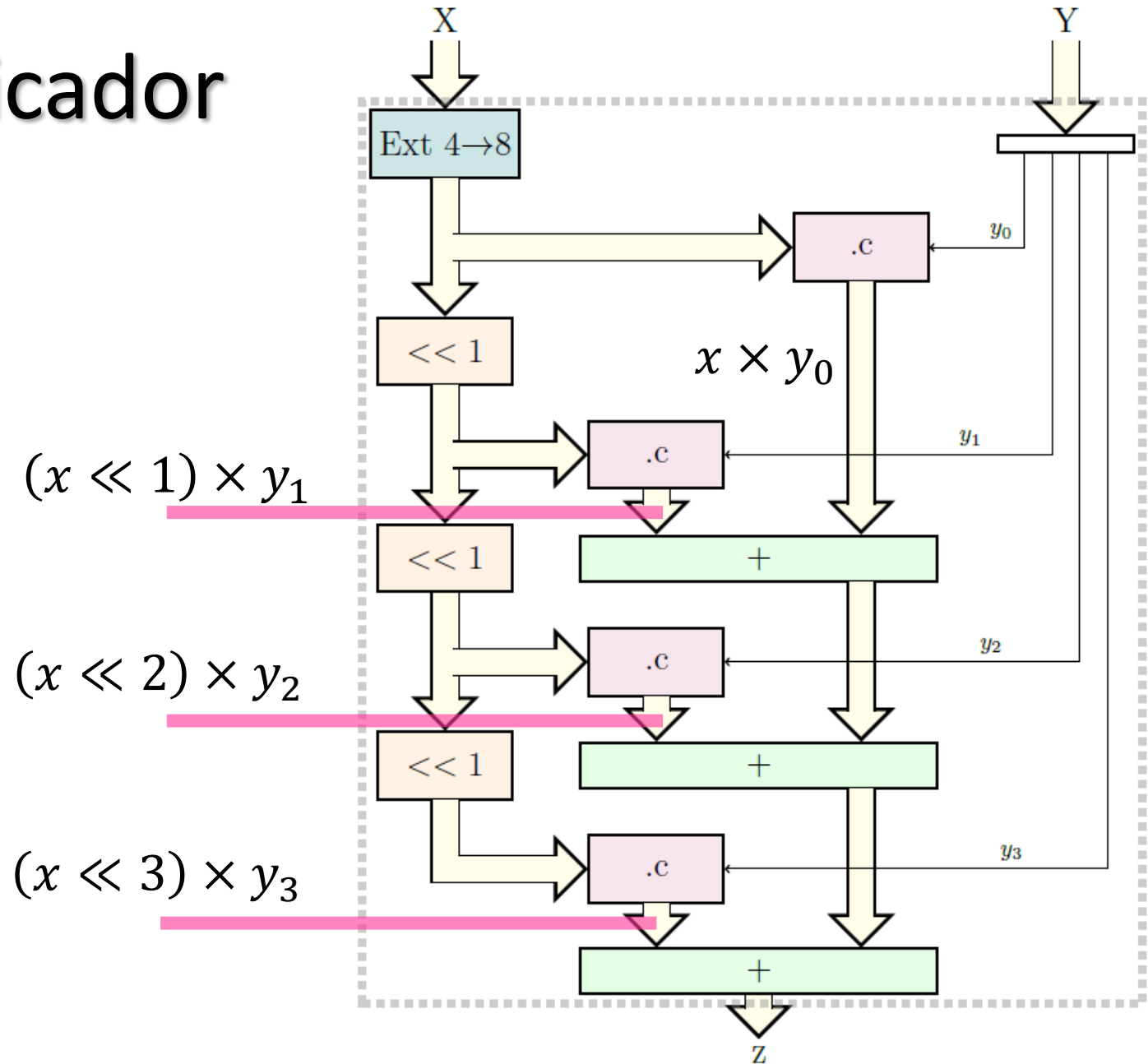
$$\begin{array}{r}
 \begin{array}{cccc} 3 & 5 & 4 & 6 \end{array} \xleftarrow{\text{blue}} x \\
 \times \begin{array}{|c|c|c|c|} \hline 1 & 4 & 5 & 7 \\ \hline \end{array} \xleftarrow{\text{blue}} y \\
 \hline
 \begin{array}{cccc} 2 & 4 & 8 & 2 & 2 \end{array} \xleftarrow{\text{orange}} x \times y_0 \\
 \begin{array}{ccccc} 1 & 7 & 7 & 3 & 0 \end{array} \xleftarrow{\text{orange}} (x \ll 1) \times y_1 \\
 \begin{array}{cccc} 1 & 4 & 1 & 8 & 4 \end{array} \xleftarrow{\text{orange}} (x \ll 2) \times y_2 \\
 \begin{array}{cccc} 3 & 5 & 4 & 6 \end{array} \xleftarrow{\text{orange}} (x \ll 3) \times y_3 \\
 \hline
 \begin{array}{cccccc} 5 & 1 & 6 & 6 & 5 & 2 & 2 \end{array}
 \end{array}$$

Para Binarios:

$$\begin{array}{r}
 \begin{array}{cccc} 1 & 1 & 1 & 0 \end{array} \xleftarrow{\text{blue}} x \\
 \times \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 1 \\ \hline \end{array} \xleftarrow{\text{blue}} y \\
 \hline
 \begin{array}{cccc} 1 & 1 & 1 & 0 \end{array} \xleftarrow{\text{orange}} x \times y_0 \\
 \begin{array}{cccc} 1 & 1 & 1 & 0 \end{array} \xleftarrow{\text{orange}} (x \ll 1) \times y_1 \\
 \begin{array}{cccc} 0 & 0 & 0 & 0 \end{array} \xleftarrow{\text{orange}} (x \ll 2) \times y_2 \\
 \begin{array}{cccc} 1 & 1 & 1 & 0 \end{array} \xleftarrow{\text{orange}} (x \ll 3) \times y_3 \\
 \hline
 \begin{array}{ccccccccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}
 \end{array}$$

En binario, el multiplicar  
por un bit es lo mismo que  
hacer and bit a bit

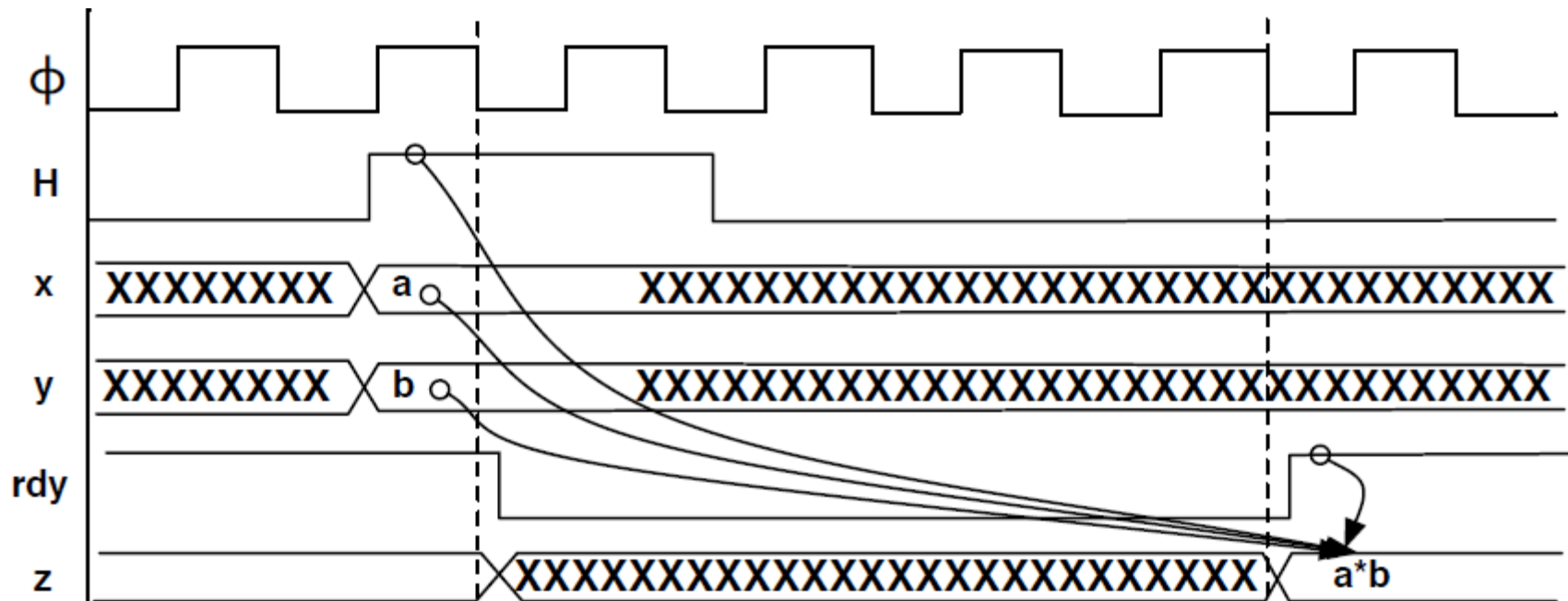
# Multiplicador 4 bits



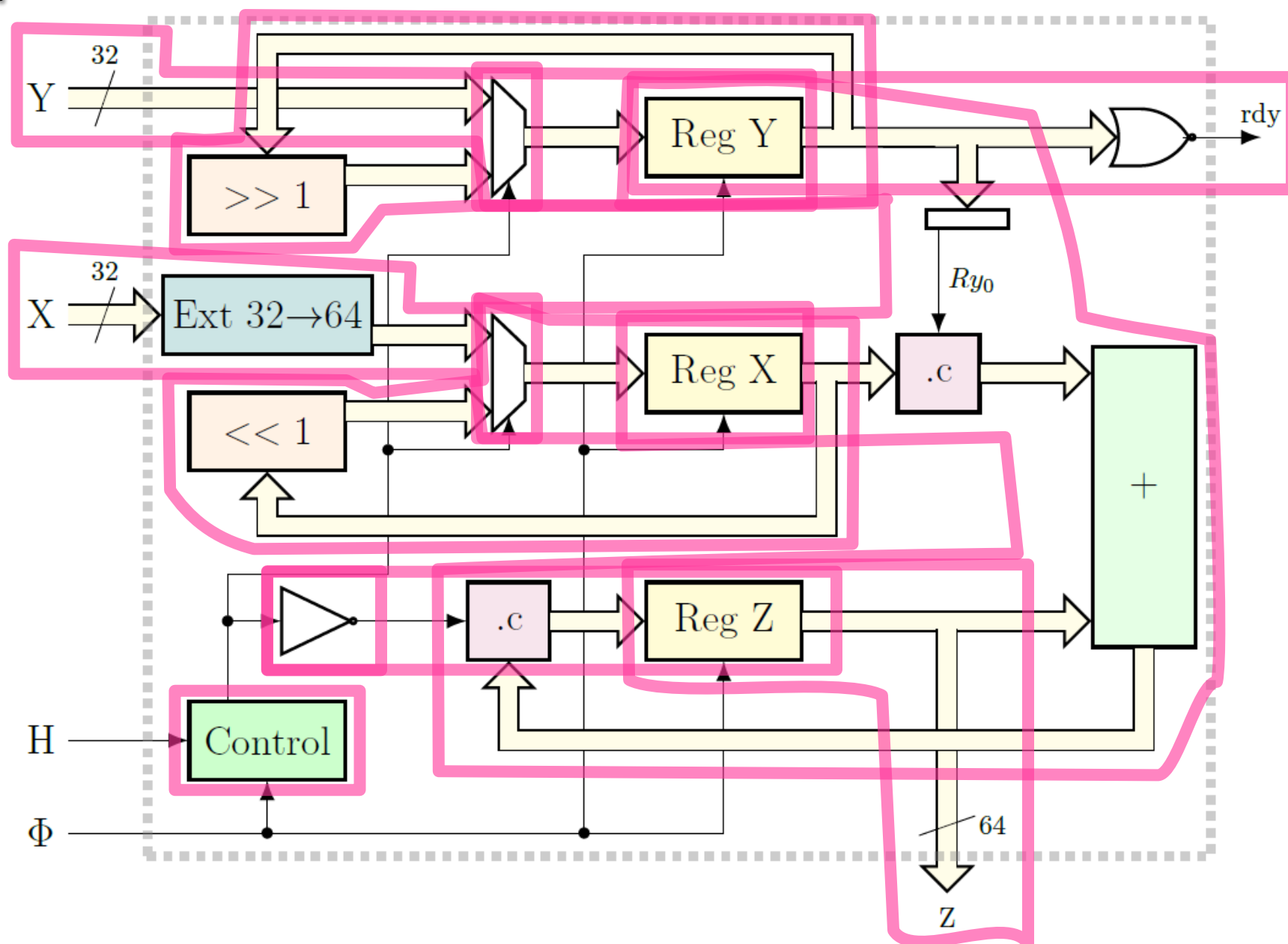
# Implementación en Serie

```
Mult (x,y de 32 bits){  
  
    sea Rx de 64 bits = Ext 32→64 (x) ;  
    sea Ry de 32 bits = y;  
    sea Rz de 64 bits = 0;  
  
    while (Ry != 0) {  
        if(Ry[0] != 0)  
            Rz += Rx;  
        Rx := Rx << 1;  
        Ry := Ry >> 1;  
    }  
  
    return Rz;  
}
```

# Diagrama de Tiempo



# Implementación





# Trabajo Grupal 2

**Table 1.2** Hexadecimal number system

| Hexadecimal Digit | Decimal Equivalent | Binary Equivalent |
|-------------------|--------------------|-------------------|
| 0                 | 0                  | 0000              |
| 1                 | 1                  | 0001              |
| 2                 | 2                  | 0010              |
| 3                 | 3                  | 0011              |
| 4                 | 4                  | 0100              |
| 5                 | 5                  | 0101              |
| 6                 | 6                  | 0110              |
| 7                 | 7                  | 0111              |
| 8                 | 8                  | 1000              |
| 9                 | 9                  | 1001              |
| A                 | 10                 | 1010              |
| B                 | 11                 | 1011              |
| C                 | 12                 | 1100              |
| D                 | 13                 | 1101              |
| E                 | 14                 | 1110              |
| F                 | 15                 | 1111              |

