OPTIMIZACIÓN DE CIRCUITOS INTEGRADOS ANALÓGICOS MEDIANTE EL USO DE SPICE

F. Serra-Graells[1], A. Uranga[2], N. Barniol[2]
[1]Centro Nacional de Microelectrónica, CSIC. paco.serra@cnm.es
[2]Dept. Ingeniería Electrónica, Universitat Autònoma de Barcelona nuria.barniol@uab.es; arantxa.uranga@uab.es
Campus UAB, 08193 Bellaterra, Spain

RESUMEN

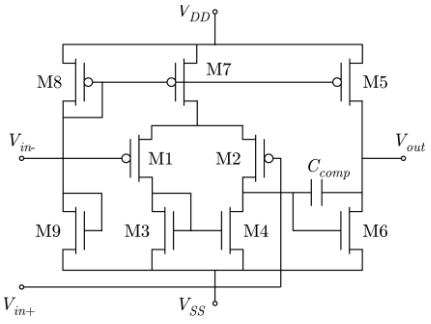
El trabajo que presentamos a continuación propone una nueva estrategia para la enseñanza de la optimización de circuitos analógicos integrados, basada en simulaciones Spice. La elección del lenguaje estándar Spice asegura la aplicabilidad de los conocimientos adquiridos en la mayor parte de los campos de diseño analógico (procesado de señal, comunicaciones de RF...). Para ilustrar esta nueva metodología, se presenta el estudio de un circuito propuesto como práctica a los alumnos.

1. INTRODUCCIÓN

El incremento de la demanda por parte del mercado actual de sistemas *on chip* (SoCs) mixtos requiere ingenieros electrónicos con un buen conocimiento tanto de circuitos analógicos como digitales. Es más, dentro de estos circuitos, el diseño de los bloques analógicos precisa más tiempo y su optimización requiere mayor esfuerzo. Con tal de facilitar el aprendizaje al alumno, se propone el uso de la herramienta CAD Spice Opus [1], suministrado gratuitamente por la Universidad de Ljubljana. Se trata de un simulador tipo Spice3 de Berkeley que proporciona además herramientas de optimización, así como la posibilidad de simulación mixta tipo XSpice [2]. Estas facilidades hacen que esta herramienta pueda también ser empleada en asignaturas de diseño *full-custom* [3] y de modelado mixto [4].

2. CIRCUITOS DE TEST

El ejemplo práctico que se presenta a continuación se basa en el análisis del típico amplificador operacional Miller de dos etapas mostrado en la Figura 1(izquierda).

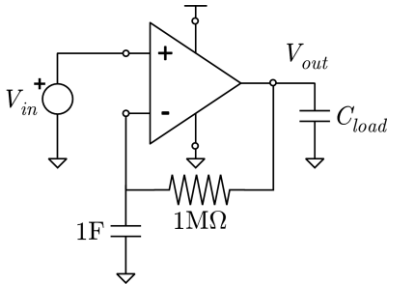


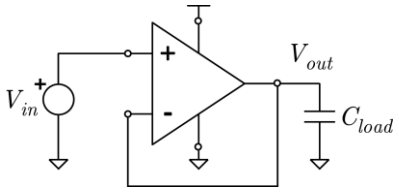

```
.subckt opamp vinn vinp vout vdd vss
m1 vload vinn vcomm vdd modp w=18u l=6u
m2 vinter vinp vcomm vdd modp w=18u l=6u
m3 vload vload vss vss modn w=12u l=12u
m4 vinter vload vss vss modn w=12u l=12u
m5 vout vbias vdd vdd modp w=96u l=6u
m6 vout vinter vss vss modn w=48u l=6u
m7 vcomm vbias vdd vdd modp w=24u l=6u
m8 vbias vbias vdd vdd modp w=12u l=6u
m9 vbias vbias vss vss modn w=3u l=24u
ccomp vout vinter cpoly w=100u l=100u
.ends
```

Fig. 1. Amplificador operacional bajo test (izquierda) y circuito equivalente Spice (derecha).

En este caso, las principales especificaciones son: ganancia DC en tensión (gdc), producto ganancia ancho de banda (gbw), margen de fase (pm), slew-rate positivo y negativo (srpos/srneg), área del circuito (area) y consumo de potencia (pd). El primer paso que ha de realizar el alumno consiste en modelar la celda a testear como un subcircuito Spice (Fig.1 derecha). Como cualquier circuito CMOS, las principales variables de diseño son las dimensiones de los transistores (W y L) y su polarización, las cuales están íntimamente relacionadas con las especificaciones de área y consumo de potencia respectivamente.

Para realizar un estudio de todas las prestaciones requeridas, el alumno debe analizar varios circuitos de test. En este caso particular, una buena solución es la configuración que se muestra en la Figura 2. La topología casi lazo-abierto (Fig. 2 izquierda) permite obtener gdc, pm y gbw, mientras que los parámetros srpos y srneg se pueden derivar de la topología de seguidor (Fig. 2 derecha). Por último, area y pd pueden ser obtenidos del análisis del punto de operación de cualquiera de los dos esquemáticos.






```
qopenloop.sp3
.include cnm25typ.mod2
.include opamp.sp3
cload vout 0 10p
vin vin 0 dc=2.5v ac=1
xopamp vfb vin vout vdd vss opamp
vss vss 0 dc=0v
vdd vdd 0 dc=5v
cdc vfb 0 1
rdc vfb vout 1000k
.end
```

```
follower.sp3
.include cnm25typ.mod2
.include opamp.sp3
cload vout 0 10p
vin vin 0 pulse(2 3 1u 1n 1n 2u)
xopamp vout vin vout vdd vss opamp
vss vss 0 dc=0v
vdd vdd 0 dc=5v
.end
```

Fig. 2: Circuitos de test: topología casi lazo-abierto (izquierda) y seguidor (derecha).

3. CARACTERIZACIÓN AUTOMÁTICA

El siguiente paso dentro del proceso de optimización es la extracción automática de las prestaciones del circuito. La misión del alumno en esta etapa del diseño consiste en la realización de distintos análisis usando diferentes circuitos de test, así como la programación de las medidas automáticas que se aplicarán a los resultados de simulación. A continuación se muestra un ejemplo:

```
.control
source qopenloop.sp3
source follower.sp3
setcirc qopenloop.sp3
op
let pd=-i(vdd)*v(vdd)*1e3
let area=(@m1:xopamp[w]+6u)*(@m1:xopamp[l]+11u)+(@m2:xopamp[w]+6u)*(@m2...
ac dec 50 10 10e6
let gmag=20*log10(mag(v(vout)))
let gph=phase(v(vout))
let gdc=gmag[0]
```

```
let gbw=abs(frequency[sum(gmag ge 0)])/1e6
let pm=180+gph[sum(gmag ge 0)]

setcirc follower.sp3
tran 1n 5u
let vrise=v(vout)*(time lt 3u)+3*(time ge 3u)
let trisebot=min(time*(vrise ge 2.1)+(vrise lt 2.1))
let trisetop=min(time*(vrise ge 2.9)+(vrise lt 2.9))
let srpos=0.8/(trisetop-trisebot)*1e-6
let vfall=v(vout)*(time ge 3u)+3*(time lt 3u)
let tfalltop=min(time*(vfall lt 2.9)+(vfall ge 2.9))
let tfallbot=min(time*(vfall lt 2.1)+(vfall ge 2.1))
let srneg=0.8/(tfallbot-tfalltop)*1e-6
.endc
```

La Figura 3 muestra tanto el *datasheet* extraído como los resultados gráficos de las simulaciones obtenidas tras la aplicación del código de caracterización anterior.

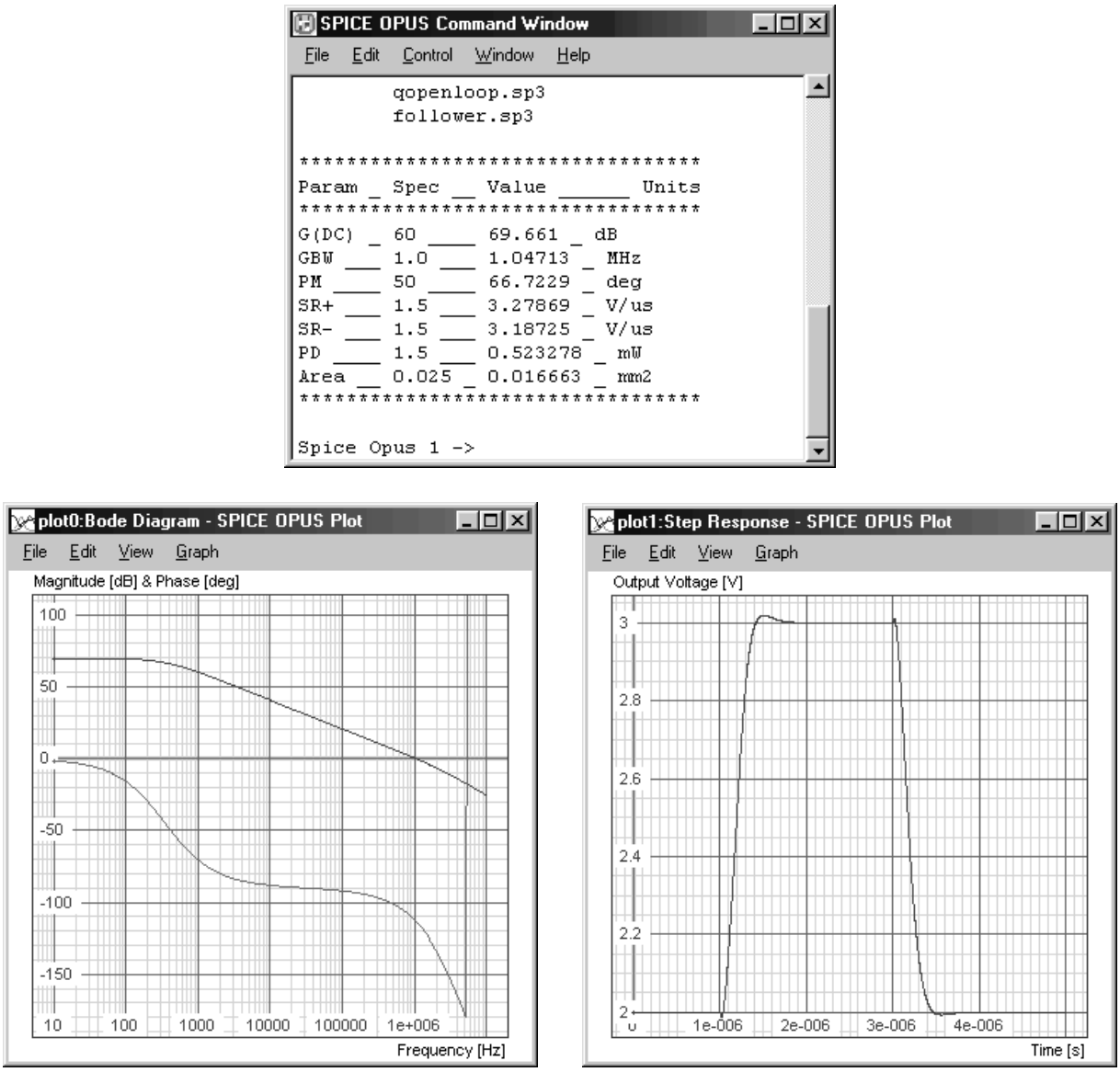


Fig. 3 *Datasheet* (arriba) y resultados de simulación (abajo) obtenidos automáticamente.

4. PROCESO DE OPTIMIZACIÓN

Una vez se consigue la evaluación automática de la celda diseñada, el estudiante comienza a definir el proceso de optimización propiamente dicho. Esta tarea implica la definición de las variables de diseño, los análisis requeridos, las restricciones y la función de coste. El estudiante debe minimizar en el mayor grado posible el número de variables de diseño. Esto es posible por ejemplo, explotando las simetrías intrínsecas del circuito. Por otro lado, a partir de las ecuaciones analíticas del diseño se pueden derivar los valores iniciales de ciertas variables así como su rango permitido de variación (valor mínimo y máximo). Para el caso del diseño tomado como ejemplo en la Figura 1 tendremos:

```
optimize parameter 0 element m1:xopamp parameter w low 12u high 200u initial 12u
optimize parameter 1 element m5:xopamp parameter w low 12u high 200u initial 96u
...
optimize analysis 0 setcirc qopenloop.sp3
optimize analysis 1 op
optimize analysis 2 let pd=-i(vdd)*v(vdd)*1e3
optimize analysis 3 let area=(@m1:xopamp[w]+6u)*(@m1:xopamp[l]+11u)+(@m2:xopamp...
...
optimize implicit 0 op1.pd lt 1.5
optimize implicit 1 op1.area lt 0.025
...
```

Finalmente, pero no por ello menos importante, el éxito de cualquier proceso de optimización no viene dado exclusivamente por el algoritmo empleado sino por la función de coste, la cual clasifica en cada iteración el resultado obtenido. Durante este proceso, los estudiantes pueden experimentar con un amplio número de expresiones y aprender de cada uno de los resultados, como se muestra en las Figuras 4 a 6.

5. CONCLUSIONES

Se ha presentado una nueva metodología de enseñanza para la optimización de circuitos integrados analógicos. Las principales ventajas que presenta son: completo control por parte del estudiante de todos los pasos del proceso, uso de un lenguaje estándar tipo Spice, alta flexibilidad, aplicabilidad a un amplio número de circuitos integrados analógicos y acceso gratuito a la herramienta de diseño CAD.

6. BIBLIOGRAFÍA

[1] Group for Computer Aided Circuit Design, "Spice Opus 2.03", University of Ljubljana, Slovenia. http://fides.fe.uni-lj.si/spice

[2] F.L. Cox *et al*, "Xspice Software User's Manual", Georgia Tech Research Corp.

[3] F. Serra-Graells and N. Barniol, "Design of Analogue Integrated Circuits: Freeware PC-based CAD for Student Practices", Microelectronics Education, Ed. Kluwer Academic Publishers, pp.289-292, 2000.

[4] F. Serra-Graells and N. Barniol, "Mixed Integrated Circuit Design with CMOS VLSI Technologies for Pre-Graduated Students", Microelectronics Education, Ed. Kluwer Academic Publishers, pp.209-212, 2002.

$$Coste = 8\left|\frac{20 - SR_-}{SR_-}\right| + 8\left|\frac{20 - SR_+}{SR_+}\right| + 10\left|SR_- - SR_+\right| + 300\left|\frac{80 - G(DC)}{G(DC)}\right| + 11\left|\frac{10^7 - GBW}{GBW}\right| + \ldots$$

```
optimize cost 8*abs((20-tran1.srneg)/tran1.srneg)+8*abs((20-tran1.srpos)/tran1.srpos)
+10*abs(tran1.srneg-tran1.srpos)+300*abs((80-ac1.gdc)/ac1.gdc)+11*abs((10e6-ac1.gbw)/ac1.gbw)
+65*abs((op1.a rea-0.025)/op1.area)+300*abs((ac1.pm-50)/ac1.pm)+250*(abs(op1.pd -1.5)/op1.pd)
```

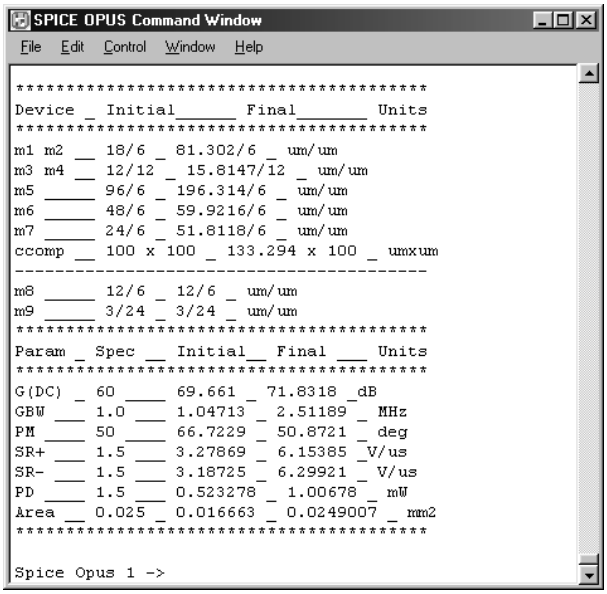


Fig. 4. Ejemplo de función de coste general y resultados.

$$Coste = \frac{1}{G(DC)}$$

```
optimize cost 1/abs(ac1.gdc)
```

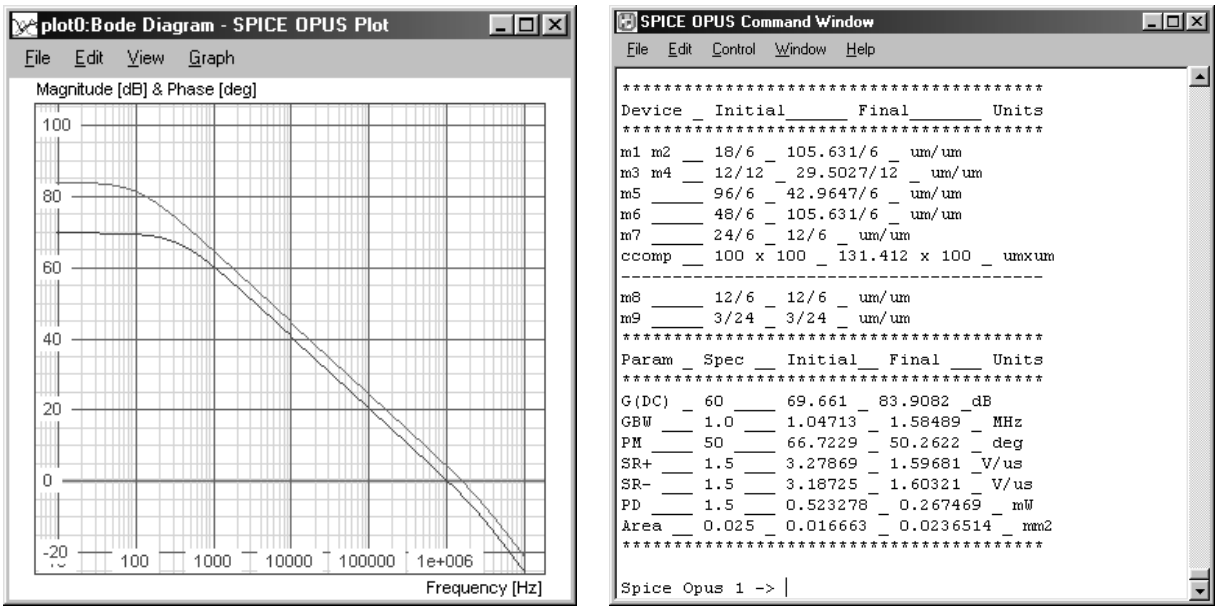

plot0:Bode Diagram - SPICE OPUS Plot
File Edit View Graph
Magnitude [dB] & Phase [deg]
100
80
60
40
20
0
-20
100
1000
10000
100000
1e+006
Frequency [Hz]
SPICE OPUS Command Window
File Edit Control Window Help
****
Device _ Initial_____ Final_____ Units
****
m1 m2 __ 18/6 _ 105.631/6 _ um/um
m3 m4 __ 12/12 _ 29.5027/12 _ um/um
m5 _____ 96/6 _ 42.9647/6 _ um/um
m6 _____ 48/6 _ 105.631/6 _ um/um
m7 _____ 24/6 _ 12/6 _ um/um
ccomp __ 100 x 100 _ 131.412 x 100 _ umxum
m8 _____ 12/6 _ 12/6 _ um/um
m9 _____ 3/24 _ 3/24 _ um/um
Param _ Spec __ Initial__ Final ___ Units
G(DC) _ 60 ____ 69.661 _ 83.9082 _dB
GBW ___ 1.0 ___ 1.04713 _ 1.58489 _ MHz
PM ____ 50 ____ 66.7229 _ 50.2622 _ deg
SR+ ___ 1.5 ___ 3.27869 _ 1.59681 _V/us
SR- ___ 1.5 ___ 3.18725 _ 1.60321 _ V/us
PD ____ 1.5 ___ 0.523278 _ 0.267469 _ mW
Area __ 0.025 _ 0.016663 _ 0.0236514 _ mm2
Spice Opus 1 ->
```

Fig. 5. Ejemplo de función de coste específica y resultados.
```

$$Coste = \frac{1}{SR_-} + \frac{1}{SR_+} + 1000 \left| P_D - 1.5 \right|$$

```
optimize cost 1/abs(tran1.srneg)+1/abs(tran1.srpos)+1000*abs(op1.pd-1.5)
```

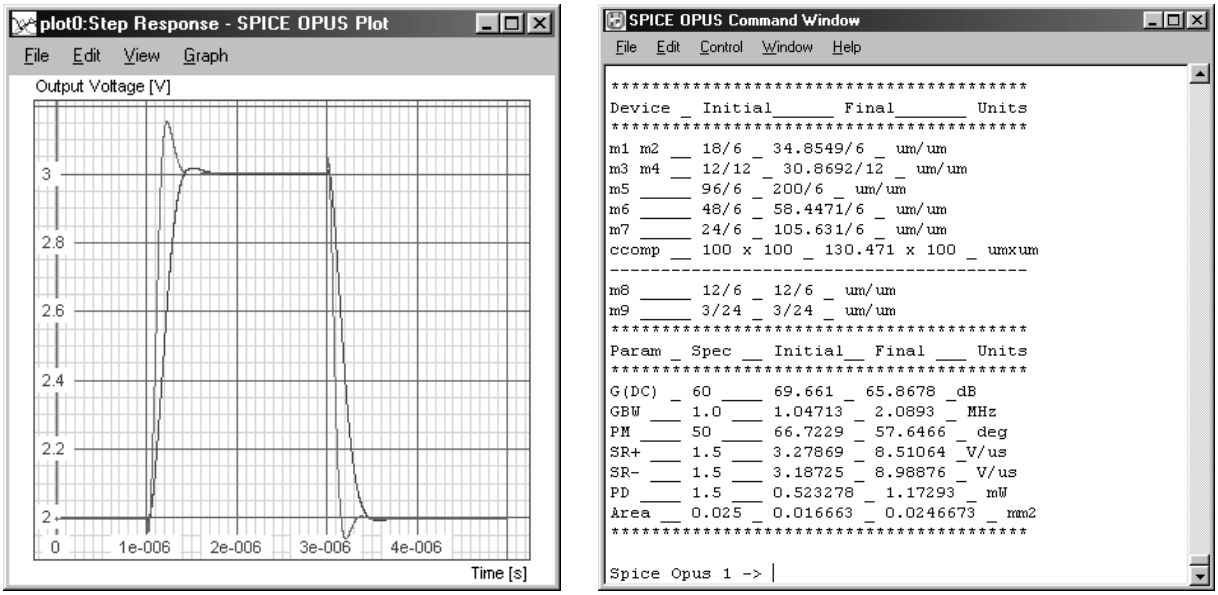


Fig. 6. Ejemplo de función de coste específica y resultados.