Samsung
EPP Demo Board Introduction

Memory Product &
Technology Division

2001.7.25

Introduction for the Demo Board System

- This application note provides example hardware interface and software for Samsung`s NAND Flash Memory

- Interfaced with the PC Parallel EPP(Enhanced Parallel Port) bus in IBM PC, Samsung NAND Flash can be programmed, erased, and read.

This Application Note :

- Shows an example of glue logic design which incorporates Samsung NAND Flash Memory
- Offers a example “C” program example to operate NAND Flash Memory.

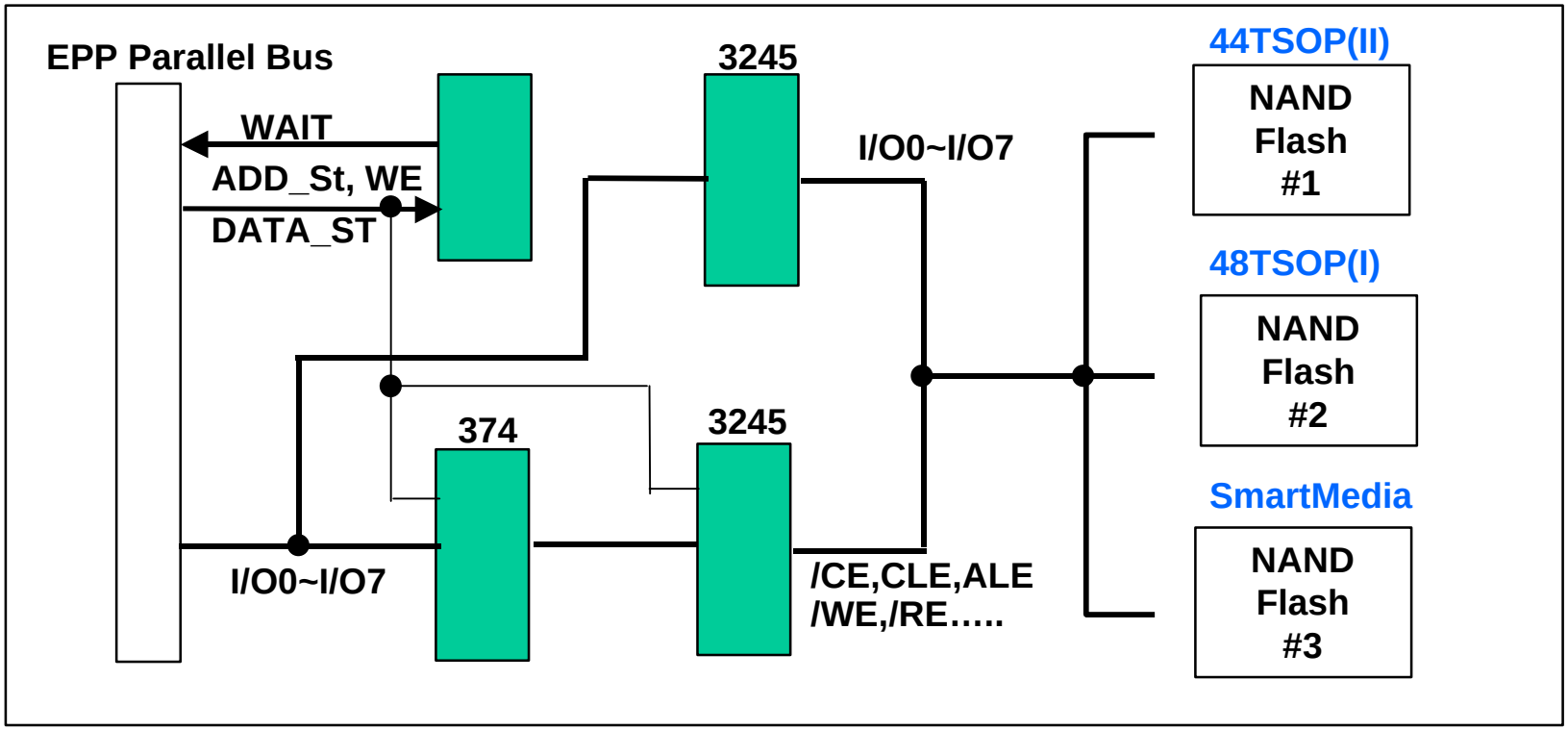
IBM PC I/O Address Map

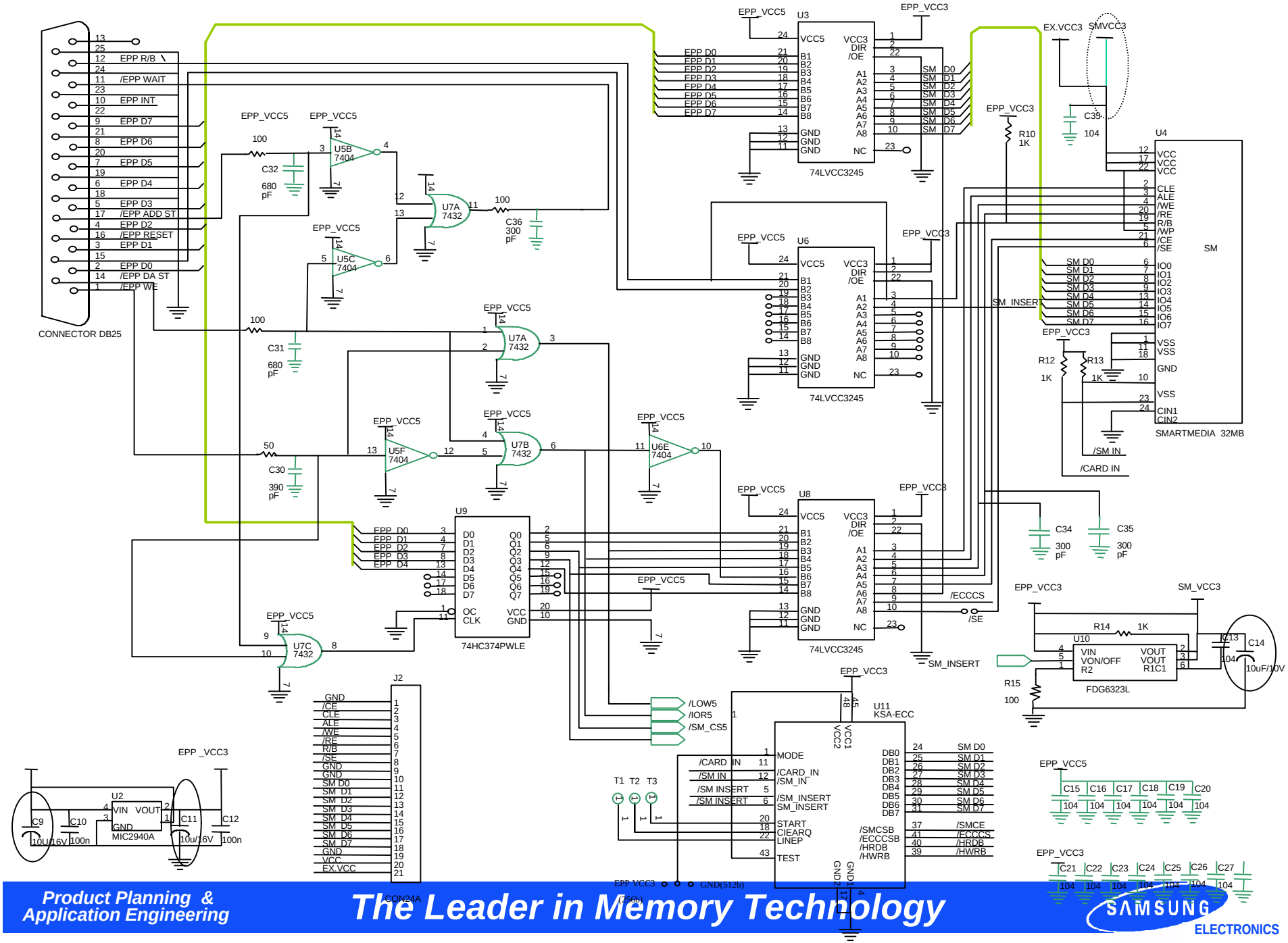
Hexa Range	Device
000 ~ 0FF	System Board I/O
1FF ~ 1F8	Fixed Disk
•	•
•	•
2F8 ~ 2FF	Serial Port 2
300 ~ 31F	*Prototype Card*
378 ~ 37F	Parallel Printer
•	•
•	•

- 378h to 37Fh are available on the Parallel Printer system, so we used 378h to 37Fh I/O port address as a data and control signal addresses.

Hardware Interfacing

- By using I/O addresses(378h~ 37bh) and six TTL chips, the NAND Flash memory can be controlled.




CONNECTOR DB25
EPP R/B
/EPP WAIT
EPP INT
EPP D7
EPP D6
EPP D5
EPP D4
EPP D3
/EPP ADD ST
EPP D2
/EPP RESET
EPP D1
EPP D0
/EPP DA ST
/EPP WE
U3
74LVCC3245
U4
SM
SMARTMEDIA 32MB
U5B 7404
U5C 7404
U5F 7404
U6E 7404
U7A 7432
U7B 7432
U7C 7432
U6
U8
74LVCC3245
U9
74HC374PWLE
U10
FDG6323L
U11
KSA-ECC
J2
CON24A
U2
MIC2940A
EPP_VCC5
EPP_VCC3
SM_VCC3
EX.VCC3
SMVCC3
SM_INSERT
/SM IN
/CARD IN
/ECCCS
/SE
/LOW5
/IOR5
/SM_CS5
/CARD_IN
/SM_IN
/SM INSERT
/SM_INSERT
SM_INSERT
T1 T2 T3
GND(512b)
GND(256b)

Operation Mode Example

```
#define SM_BASE   0x378    // EPP Address

uint g_smStatAddr = SM_BASE + 1;
uint g_smCtrlAddr = SM_BASE + 2;
uint g_smAddrAddr = SM_BASE + 3;
uint g_smDataAddr = SM_BASE + 4;

#define DATA          SM_BASE+4
#define CTL           SM_BASE+4

#define add_latch_enable(){  \
     outportb(g_smCtrlAddr, 4); \
     outportb(g_smAddrAddr,0x0a);  \
       }
#define add_latch_disable(){ \
       outportb(g_smAddrAddr, 0x08); \
       outportb(g_smCtrlAddr, 0); \
       }
#define cmd_latch_enable(){ \
       outportb(g_smCtrlAddr,4); \
       outportb(g_smAddrAddr, 0x09); \
       }
```

```
#define cmd_latch_disable(){ \
       outportb(g_smAddrAddr, 0x08); \
       outportb(g_smCtrlAddr, 0); \
       }
#define chip_enable(chip_no){ \
       outportb(g_smCtrlAddr,4); \
       switch(chip_no){ \
     case 1: \
       outportb(g_smAddrAddr, 0x08); \
       break; \
     case 2: \
       outportb(g_smAddrAddr, 0x08); \
       break; \
       } \
       }
#define chip_disable(){  \
       outportb(g_smAddrAddr, 0x0c); \
       outportb(g_smCtrlAddr, 0); \
       }
```

Operation Mode Example(Continue)

```
#define ecc_enable(){ \
        outportb(g_smCtrlAddr,4); \
        outportb(g_smAddrAddr, 0x04); \
        }
#define ecc_disable(){ \
        outportb(g_smAddrAddr, 0x0c); \
        outportb(g_smCtrlAddr, 0); \
        }
#define sm_write_en() \
        outportb(g_smCtrlAddr,0x4)

#define sm_write_dis() \
        outportb(g_smCtrlAddr,0x0)

#define sm_read_en()  \
        outportb(g_smCtrlAddr, 0xf4)

#define sm_read_dis()  \
      outportb(g_smCtrlAddr, 0)
```