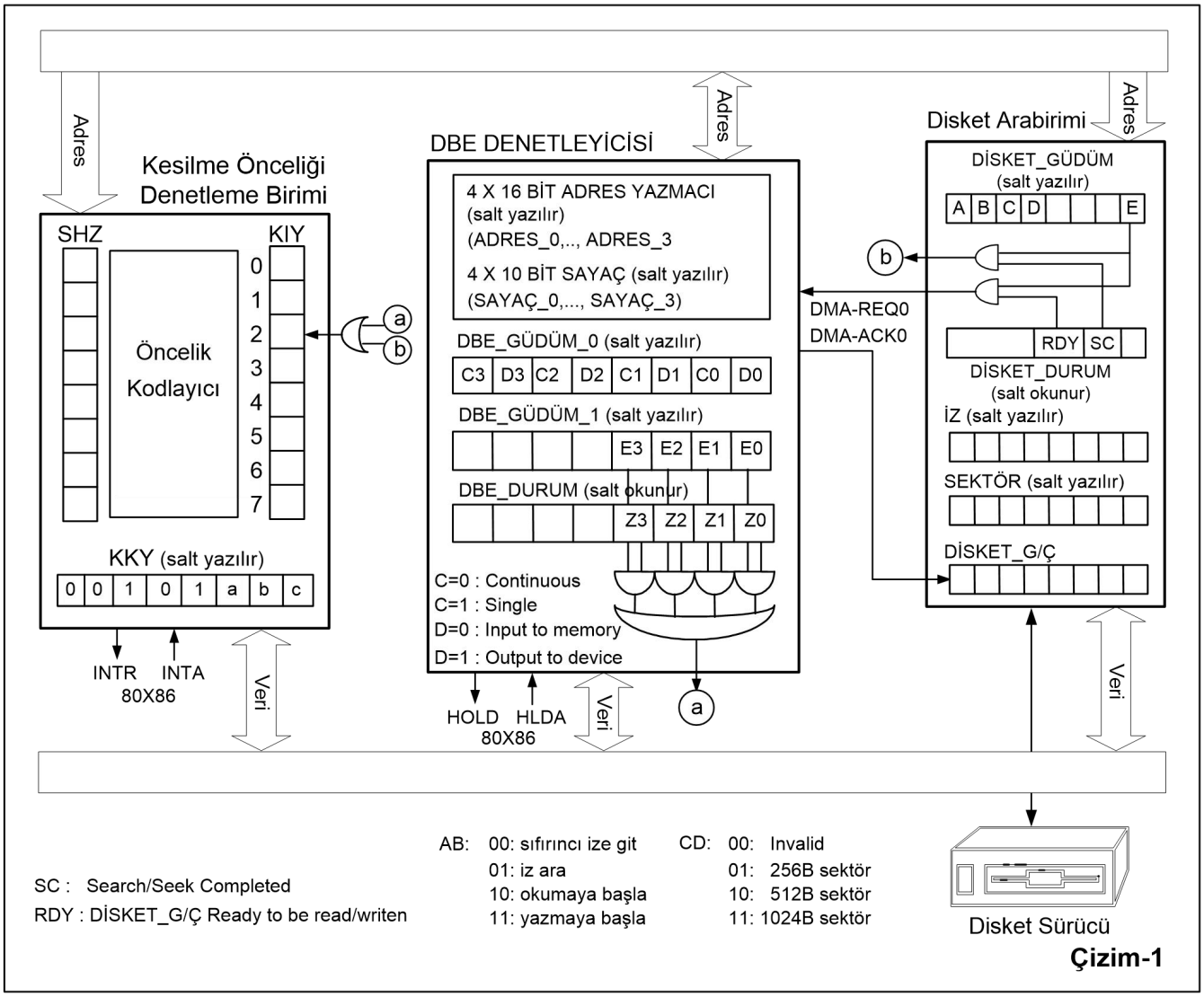
Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü
BİL323 İşletim Dizgeleri I

Genel Sınav (27/01/2006)
Prof.Dr. Ali SAATÇİ

Öğrencinin
Adı Soyadı :



Çizim-1

Soru-1. Çizim-1'de verilen konfigürasyonu taban alarak Disket Sürücü üzerinde `{İZ_0-SEKTÖR_0}` ikilisi ile tanımlanan 512 baytlık sektör içeriğinin, `{İZ_1-SEKTÖR_1}` ikilisi ile tanımlanan sektöre yazılması için gerekli başlatma kesimi ile kesilme yordamını veriniz (`baslatma`, `kesilme`).

Bunu yaparken:

a. Arabirim ve yardımcı işleyiciler üzerindeki yazmaçların fiziksel adresleri olarak, Çizim-1'de verilen (`SHZ`, `DİSKET_GÜDÜM`, `ADRES_0` gibi) yazmaç adlarını imge olarak kullanınız.

b. Yazmaç içerik ve kimi değişmez değerleri olarak, imgeler yerine, yerine göre, ikili, onlu, onaltılı sayısal değerlere doğrudan yer veriniz.

c. Ana bellekte aktarım yapılacak yastık alan adı olarak `disk-buffer` imgesini kullanınız.

d. Veri kesim (`data segment`) tanımına yer vermeniz istenmemektedir.

Soru-2. Aşağıda, *UNIX* ortamında, birlikte çalışan işletim dizilerine ilişkin, **Pthreads** kitaplığında yer alan kimi işlevler verilmiştir.

a. Önce, işletim sistemleri, birlikte çalışan görevler konusu kapsamında öğrendiklerinizden yararlanarak bu işlevlerin nasıl ve ne amaçla kullanılacağına ilişkin bir görüş oluşturunuz. Bu bağlamda, **pthread_cond_wait()** ve **pthread_cond_signal()** işlevlerinin kullanım amaçlarını, monitor konusunda öğrendiklerinizden de yararlanarak açıklayınız. Bunu yaparken **pthread_cond_wait()** işlevinin argümanları arasında **mutex** değişkenine niye yer verilmiş olabileceğini açıklamayı unutmayınız.

b. Sonra, verilen son dört işlevi de kullanarak, N bayttan oluşan bir yastık üzerinden, üretici tüketici ilişkisi içinde bayt alış-verişi gerçekleştiren iki işletim dizisini C ile programlayınız.

`üretici`'nin yastığa yazacağı baytı klavyeden okuyacağı; `tüketici`'nin de yastıktan okuduğu baytı ekrana yazdıracağı varsayılacaktır.

Programınızda işletim dizisi yaratma-yoketme; koşul değişkeni yaratma-yoketme işlemlerine yer vermeniz istenmemektedir. Aşağıda, bu amaçlara dönük işlevler, diğer işlevlerin anlamlarını tamamlamak/pekiştirmek için verilmiştir.

```
int pthread_mutex_init (pthread_mutex_t *mut, NULL);
int pthread_mutex_destroy (pthread_mutex_t *mut);
int pthread_cond_init (pthread_cond_t *cond, NULL);
int pthread_cond_destroy (pthread_cond_t *cond);

int pthread_mutex_lock (pthread_mutex_t *mut);
int pthread_mutex_unlock (pthread_mutex_t *mut);
int pthread_cond_wait (pthread_cond_t *cond, pthread_mutex_t *mut);
int pthread_cond_signal (pthread_cond_t *cond);
```