



Araştırma Makalesi

## Gerçek Zamanlı Gömülü Sistemlerde Enerji Tüketiminin Azaltılması İçin Teknikler

Abdullah ELEWİ\*<sup>1</sup>, Ayşegül YAMAN<sup>1</sup>, Sibel KAPLAN<sup>1</sup>, Ahmed Abd ALKADER<sup>2</sup>

<sup>1</sup>Mersin Üniversitesi, Bilgisayar Mühendisliği Bölümü, Mersin, Türkiye

<sup>2</sup>Hatay Mustafa Kemal Üniversitesi, Hassa Meslek Yüksekokulu, Hatay, Türkiye

### ÖZ

**Anahtar Kelimeler:**  
Gerçek zamanlı sistem  
Enerji tüketimi  
DVFS  
DPM  
MCRTsim

Gerçek zamanlı gömülü sistemler, basit gadget'lardan karmaşık aviyonik sistemlere kadar modern hayatımızın her detayına dahil edilmiştir. Güç, gerçek zamanlı gömülü sistemlerin tasarımı ve işletiminde optimizasyon için önemli ölçütlerden biridir. İşlemci biriminde enerji tüketimini azaltmak için başlıca kullanılan teknikler; işlemcinin yavaşlama faktörlerine dayalı dinamik voltaj/frekans ölçeklendirmesi (DVFS) ve dinamik güç yönetimi (DPM)dir. Bu çalışmada kaynaklara erişim protokolleri ve gerçek zamanlı zamanlama algoritmaları kullanarak; DVFS içerisinde yer alan MaxSpeed (MS) ve kritik bölüm maksimum hız (CSMS) teknikleri, ve DPM incelenmiş ve yapılan örnekler MCRTsim programı üzerinde çalıştırılıp sonuçları alınıp karşılaştırılmıştır. Elde edilen sonuçlar, bu tekniklerin uygulanmasının, kullanılan örneklerde enerji tüketimini %27'ye kadar azaltabileceğini göstermiştir.

## Energy Consumption Reduction Techniques in Real-Time Embedded Systems

### ABSTRACT

**Keywords:**  
Real-time system  
Energy consumption  
DVFS  
DPM  
MCRTsim

Real-time embedded systems are incorporated in every detail of our contemporary life from simple gadgets to avionics complex systems. Power is one of the important criteria for optimization in the design and operation of real-time embedded systems. Mainly used techniques to reduce energy consumption in the processor unit are dynamic voltage/frequency scaling (DVFS), based on processor slowdown capabilities, and dynamic power management (DPM). In this study, using resource access protocols and real-time scheduling algorithms; MaxSpeed (MS) and critical section maximum speed (CSMS) techniques of DVFS and DPM were examined with examples run on the MCRTsim simulation program to obtain and compare results. The obtained results showed that applying such techniques can reduce energy consumption to ratios of 27% for the utilized examples.

\* Sorumlu Yazar

(elewi@mersin.edu.tr) ORCID ID 0000 - 0001 - 9774 - 5292  
(aysegulyaman@mersin.edu.tr) ORCID ID 0000 - 0002 - 6972 - 6657  
(sibelkaplan@mersin.edu.tr) ORCID ID 0000 - 0003 - 3299 - 4882  
(akader@mku.edu.tr) ORCID ID 0000-0002-0538-7924

## 1. GİRİŞ

Gömülü bir sistem, bağımsız bir sistem olarak veya büyük bir sistemin parçası olarak özel bir işlevi yerine getirmek üzere tasarlanmış yazılıma sahip mikroişlemci tabanlı bir bilgisayar donanım sistemidir. Çekirdekte, gerçek zamanlı işlemler için hesaplama yapmak üzere tasarlanmış entegre bir devre bulunur. Karmaşıklıklar, tek bir mikro denetleyiciden bağlı çevre birimleri ve ağları olan bir işlemci paketine kadar uzanır. Gömülü bir sistemin karmaşıklığı, tasarlandığı göreve bağlı olarak önemli ölçüde değişir (Lipskoch ve Ark., 2007).

Gerçek zamanlı işletim sistemleri genellikle, gömülü sistemlerde bulunur ve çalışmasında zaman kavramının çok önemli olduğu yerlerde kullanılır. Elektronik bir cihazın kontrol kartında bulunan ve sistem içerisinde görev alan yapıların belli bir işlem sırası vardır. Gerçek zamanlı işletim sistemleri, gerçek zaman ve işletim sistemi olarak iki parçadan oluşur. Bu sistemler, çoklu görev bilincinde çalışırlar. İşlemler, yapılarında bulunan bir çekirdek üzerinden gerçekleştirilir. Normal işletim sistemlerinden farkı, işlem önceliği yerine zaman önceliğine sahip olmalarıdır. Kendisine verilen görevleri, birbirleri arasında çok hızlı bir şekilde geçiş sağlayarak yerine getirerek, bütün işlemler aynı anda gerçekleşiyor izlenimi verirler. Gerçek zamanlı bir sistem, iyi tanımlanmış, sabit zaman kısıtlamaları olan zamana bağlı bir sistemdir (Liu, 2000; Laplante ve Ovaska, 2012).

Gerçek zamanlı gömülü sistemlerde güç tüketiminde enerji tasarrufu dikkate alınması gereken önemli bir faktördür. Bu sistemlerde güç tüketimini azaltmanın iki ana yolu vardır: işlemcinin yavaşlama faktörlerine dayalı dinamik voltaj ve frekans ölçeklendirmesi (DVFS) ve dinamik güç yönetimi (DPM)dir (Schmitz ve Ark., 2005). Frekans veya voltaj ölçeklendirme kullanarak yavaşlatma, güç tüketimini azaltmada daha etkilidir. Bir

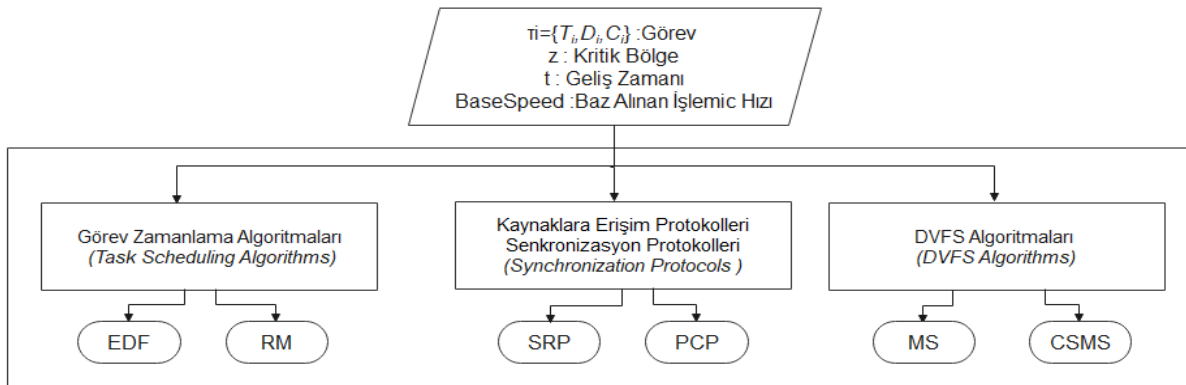
işlemcinin frekansını ve voltajını ölçeklendirmek, bir işin yürütme süresinde bir artışa yol açar. Enerjiyi en aza indirme hedefimize ulaşmak için zamanı ve gücü mantıklı bir şekilde yönetmek zorundayız.

DVFS, besleme voltajını ve çalışma frekansını düşürerek enerji kaybını azaltmak için işlemcilerin donanım özelliklerinden yararlanmada önemli bir teknik olmuştur. DVFS algoritmalarının, genel amaçlı sistemlerde en yüksek işlemci gücünde çalışırken, önemli ölçüde enerji tasarrufu sağlayabildiği gösterilmiştir (Pillai ve Shin, 2001; Awadalla ve Elewi, 2016; Saad ve Ark., 2013).

Bu yazıda, sabit yavaşlama faktörlerinin hesaplanması yoluyla sistem düzeyinde güç yönetimine odaklanılmıştır. Literatürde birden fazla DVFS algoritması mevcuttur. Bu çalışmada; DVFS algoritmalarından MS (MaxSpeed) ve CSMS (Critical Section Maximum Speed) algoritmaları incelenmiştir. Bu algoritmalar incelenirken, zamanlama algoritması olarak Monoton Oran (Rate Monotonic-RM) ve Erken Biten Önce (Earliest Deadline First-EDF) kullanılmıştır (Palamut ve Ark., 2019; Yıldırım ve Ark., 2020). Paylaşılan kaynaklara erişmek için Yığın Kaynak Politikası öncelikli tavan protokolü (Priority Ceiling Protocol - PCP) (Cheng ve Ras, 2007) ve (Stack-Resource Policy - SRP) (Baker, 1991) algoritmaları kullanılmıştır. Sistem tasarımı için yapılan örnekler MCRTsim (Wu ve Huang, 2017) programı üzerinde çalıştırılıp sonuçları alınmıştır. MCRTsim programı java tabanlı ve açık kaynaklı bir uygulamadır.

## 2. SİSTEM İÇİN GEREKEN BİLEŞENLER

Bu bölümde, tek işlemcili gerçek zamanlı sistem için gerekli girdi parametrelerini ve uygulama için kullanılacak algoritmaları tanımlıyoruz ve formüle ediyoruz. Şekil 1 bu bileşenleri özetlemektedir.



Şekil 1: Tek İşlemcili Gerçek Zamanlı Sistem İçin Gereken Bileşenler

### 2.1. Sistem Girdileri

Sistem modeli için öncelikli olarak gerçek zamanlı işletim sisteminde kullanılan bazı terimlerin karşılığına bakmalıyız.

- $\tau$  Görev (Task): Sistem işlevlerini birlikte sağlayabilen bir dizi
- İş (Job): İşlemciye atanan küçük iş parçası
- $r$  Görevin geliş zamanı (arrival time)
- $T$  (Periyod): Görevin periyodu

- $D$  İşin Zaman Sınırı (*Relative deadline*): İşin tamamlanması gereken zaman
- $C$  En Kötü Çalışma Süresi (*Worst case execution time: WCET*): İşin bitmesi için geçen süre
- $z$  Kritik Bölüm (*Critical Section*): Eşzamanlı çalışan ve aynı kaynağa ihtiyaç duyan iki görevden birinin kaynağı işi bitene kadar erişimi kilitlediği zaman aralığıdır.
- $n$  Görev Sayısı
- $U$  İşlemci Kullanım Yüzdesi
- $S$  Semafor

Bir görev seti  $\Gamma = \{\tau_1, \tau_2, \tau_3, \dots, \tau_n\}$  şeklinde görevlerden oluşur ve her bir görev ise  $\tau_i = \{T_i, D_i, C_i\}$  şeklinde parametrelere sahiptir.  $T_i$  görevin periyodu,  $D_i$  görevin zaman sınırı,  $C_i$  ise çalışma süresi (Worst case execution time: WCET) parametrelerini ifade etmektedir. ( $D_i \leq T_i$  olmalıdır). Görevin her çağrılışına iş (*Job*) denir. Bir görev setinin gerçekleştirilebilmesi için görevlerin her birinin kendi zaman sınırında tamamlanması gereklidir. Bunun için  $U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1$  formülü sağlanmalıdır.

Sistem, görevler tarafından karşılıklı ayrıcalıklı bir şekilde erişilen bir dizi paylaşılan kaynağa sahiptir. Paylaşılan bir kaynağa erişim izni verilen görev, kritik bölümde yürütülür.  $\tau_i$  görevinin  $k$ 'inci kritik bölümü  $z_{i,k}$  olarak temsil edilir. Paylaşılan bir kaynağı kullanması gereken bir görev, o kaynak daha düşük öncelikli bir görev tarafından kullanılıyorsa kaynak serbest bırakılana kadar engellenir. Kaynağı tutan göreve Engellenen Görev (*Blocking Task*) denir (Jejurikar ve Gupta, 2002; Jejurikar ve Gupta, 2006).

Bir görevin engellendiği süre Görev Engelleme Süresi (*Task Blocking Time*) olarak adlandırılır. Belirtilen görev bilgileri ve belirli bir kaynak erişim protokolü ile bir görev için maksimum engelleme süresi hesaplanabilir.  $B_i$ , belirli bir kaynak erişim protokolü altında  $\tau_i$  görevi için maksimum engelleme süresidir (Jejurikar ve Gupta, 2002).

## 2.2. Görev Zamanlama Algoritmaları (Task Scheduling Algorithms)

Gerçek zamanlı gömülü bir sistemde DPM/DVFS'nin enerji tüketimi azaltmak, faydalarını gerçekleştirmek için işletim sisteminin görev zamanlama algoritması ile sıkı bir şekilde bağlantılı olması gerekir. 1973 yılında Liu ve Layland (1973), optimal dinamik öncelik zamanlama algoritması olarak Erken Biten Önce (Earliest Deadline First-EDF) algoritması ve optimal sabit öncelik zamanlama algoritması olarak Monoton Oran (Rate Monotonic-RM) algoritmasını sunmuştur. RM algoritmasında görev önceliği görev süresinin uzunluğuna göre verilir. Kısa süreli görevler daha yüksek çalışma önceliğine sahiptir. EDF, görevleri son teslim tarihlerine göre sıralayan dinamik bir öncelik zamanlayıcıdır.

Gerçek zamanlı sistemde bir görev setinin uygun bir şekilde zamanlaması için aşağıdaki koşulları sağlaması gerekir (Liu ve Layland, 1973).

$$\text{EDF için } U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1 \quad (1)$$

$$\text{RM için } U = \sum_{i=1}^n \frac{C_i}{T_i} \leq n(\sqrt{2} - 1) \quad (2)$$

## 2.3. Kaynak Erişim Protokolleri (Resource Access Protocols)

Kaynak erişim protokolleri birden fazla görevin doğru çalışabilmesi için aynı kaynak veya kaynaklara ihtiyaç duyması halinde yaşanan sorunları çözmek için geliştirilmiş tekniklerdir. Protokolün temeli olan "Öncelik Değiştirme Yaklaşımı"na dayanmaktadır (Palamut ve Ark., 2019). Çalışmamızda Öncelikli Tavan Protokolü (Priority Ceiling Protocol-PCP) ve Yığın-Kaynak Politikası (Stack-Resource Policy -SRP) protokolleri kullanılmıştır.

Öncelikli Tavan Protokolü (PCP), Öncelik Kalıtım Protokolü (Priority Inheritance Protocol-PIP)' ye benzer ve aynı zamanda önleyici zamanlamaya dayalıdır. PCP ayrıca aşağıda istenen özellikler: (1) kilitlenmeleri önler; ve (2), zincirleme engellemeyi önler, böylece yüksek öncelikli bir görev, kritik bölümde kendini askıya alsa bile, en fazla bir düşük öncelikli görev tarafından engellenebilir. PCP, PIP'nin neden olduğu doğrudan engelleme ve doğrudan engellemeye ek olarak üçüncü bir engelleme türü olan tavan engellemeyi sunar. Kilitlenmelerin (deadlocks) ve zincirleme blokajların önlenmesi için tavan blokajı gereklidir (Cheng ve Ras, 2007; Buttazzo, 2011).

PCP'nin temel önermesi aşağıdaki kurallara dayanmaktadır: (1) daha yüksek öncelikli bir  $\tau$  görevini engelleyen daha düşük öncelikli bir görev,  $\tau$  görevinin önceliğini devralır ve yalnızca görev, sahip olduğu semaforları serbest bıraktığında önceliği normal değerine geri yüklenir. (2) Bir  $\tau$  görevi yalnızca bir  $S$  semaforunu kilitleyebilir: (a)  $S$  semaforu henüz kilitlenmemişse ve (b)  $\tau$  görevinin önceliği tüm semaforların öncelik tavanlarından daha büyükse. Semaforlar şu anda  $\tau$  görevi dışındaki görevler tarafından kilitlenmiştir. Bir semaforun öncelik tavanı, o semaforu herhangi bir zamanda kilitlemek isteyenlerin en yüksek öncelikli görevi olarak tanımlanır (Sha ve Goodenough, 1990).

Yığın-Kaynak Politikası (Stack-Resource Policy -SRP) Baker (1991) tarafından ortaya sunulan bir tekniktir. Paylaşılan kaynaklar erişmek için sunulmuştur. Öncelikli tavan protokolüne (PCP) ek olarak; çok birimli kaynakların kullanımına izin verir, Dinamik öncelik zamanlamasını destekler ve çalışma zamanı yığın tabanlı kaynakların paylaşımına izin verir (Buttazzo, 2011).

## 2.4. Enerji Tüketiminin Azaltılması Teknikleri

İşlemcilerin hızı arttıkça enerji tüketimi de arttı. Bir işlemcinin enerji tüketimini azaltmak önemli bir araştırma konusu haline geldi. Bu amaçla DPM ve DVFS teknikleri geliştirilmiştir.

İşlemci için güç tüketimi denklem (3)'te modellenmiştir (Wu ve Huang, 2017).

$$P(f) = \alpha f^\gamma + \beta \quad (3)$$

Burada  $P$  güç,  $f$  frekans (hız),  $\alpha$ ,  $\beta$ , ve  $\gamma$  işlemci ile ilgili sabitlerdir. Örneğin, Marvell Xscale PCA 270 işlemci için  $\alpha = 1.52$ ,  $\beta = 0.08$  ve  $\gamma = 3$ 'tür (Wu ve Huang, 2017). Daha sonra, T zaman periyodundaki enerji tüketimi, güç tüketiminin T ile çarpımı olarak basitçe modellenilebilir.

DPM, işlemci boştayken basitçe kapatmak anlamına gelir. DVFS algoritmalarında işlemcinin her bir çekirdeğinin var olan en düşük ve en yüksek hızları arasında çalıştığı kabul edilir.

Birçok DVFS algoritması mevcuttur. DVFS algoritmalarından biri olan MaxSpeed (MS) algoritmasında işlemci tüm görevler için en yüksek hızda çalışır. Kritik Bölümde En Yüksek Hız (Critical Section Maximum Speed -CSMS) Algoritmasında görevlerin kritik olan ve kritik olmayan bölgelerinin farklı hızlarda çalıştırılması amaçlanmıştır. Bütün görevlerin kritik bölümleri işlemcinin en yüksek hızında çalıştırılmakta iken kritik olmayan bölgeler için yavaşlama faktörleri hesaplanmaktadır. Yani bu algoritmada iki farklı hız kullanılır (Maximum Speed ve baseSpeed). Burada önemli olan tüm görevlerin son teslim süresinden önce sonlandırılmasıdır (Jejurikar ve Gupta, 2002).

### 3. ENERJİ TÜKETİMİNİN AZALTILMASI TEKNİKLERİNİN MCRTsim SIMÜLYONU ÜZERİNDE UYGULAMA ANALİZİ

DVFS bir enerji tasarrufu tekniğidir, bu özelliği besleme voltajını ve çalışma frekansını düşürerek çekirdeğin enerji kaybını azaltarak elde eder.

MCRTsim, tek işlemcili, çok işlemcili ve çok çekirdekli işlemcili gerçek zamanlı sistemler için açık kaynaklı bir görev zamanlama simülatörü olarak sunulmuştur (Wu ve Huang, 2017). MCRTsim kullanarak mevcut zamanlama algoritmalarının yanı sıra senkronizasyon protokollerinin performansını kolayca değerlendirebiliriz. Ayrıca, MCRTsim, minimum çabayla yeni zamanlama ve senkronizasyon protokollerinin tasarımını desteklemek için bir Java sınıf kitaplığı da içerir. MCRTsim'in bir diğer önemli özelliği, DVFS etkin işlemcilerin desteklenmesidir, böylece enerji duyarlı zamanlama algoritmalarının ve senkronizasyon protokollerinin yeteneklerinin daha iyi anlaşılması sağlanır.

Bu bölümde MCRTsim simülatör üzerinde, zamanlama algoritmaları olarak RM/EDF ve kaynak erişim protokolleri olarak da PCP/SRP algoritmaları ile birlikte DVFS/DPM tekniklerinin kullanımları örneklendirilmiş ve algoritmalar enerji verimliliği açısından analiz edilmiştir. Örneklendirmeler için bir paylaşılan kaynak ve iki periyodik görevden oluşan görev seti kullanılmıştır (Jejurikar ve Gupta, 2002). Aşağıdaki çalıştırılan tüm uygulamalarda bu tasarım kullanılmıştır.

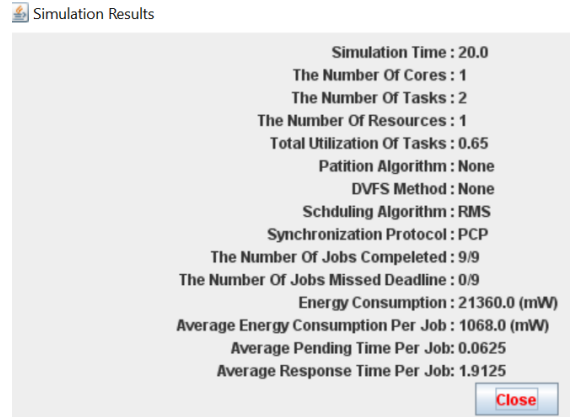
$$\tau_i = \{r_i, T_i, D_i, C_i\}$$

$$\tau_1 = \{0.5, 4, 4, 1\} \quad \tau_2 = \{0, 5, 5, 2\}$$

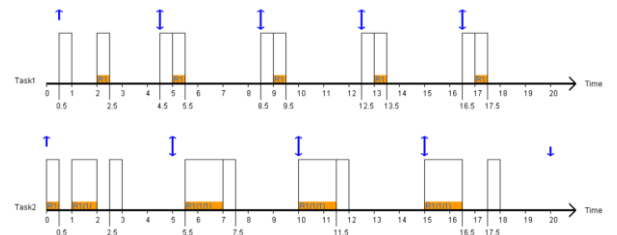
$\tau_1$  görevi için kritik bölüm  $[0.5, 1]$  ve  $\tau_2$  için  $[0, 1.5]$  değerindedir. Kritik bölümler görevleri bloke edebilir ve görevler için maksimum engelleme süresi max hızda  $B_1 = 1.5$  ve  $B_2 = 0$ 'dır. İşlemcinin maksimum tam hızı (%100) 1000 birim, ve  $\{0, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$  birim hızlar çalışabilir. Bu örnek için EDF kullanarak ve hiç ortak kaynak kullanmadan, yavaşlama (slowdown) faktörü  $U = (2/5) + (1/4) = 0.65$  şeklinde hesaplanır.

#### 3.1. DPM/DVFS metodunu RM ve PCP ile Uygulama

MCRTsim üzerinde verilen konfigürasyon için zamanlama algoritması EDF, DVFS metodu kullanılmadan, kaynak erişim protokolü SRP olmak üzere elde edilen simülasyon sonuçları ve zamanlama grafikleri Şekil 2-a, Şekil 2-b ve Şekil 2-c de verilmiştir. Şekil 2-b, her görevi tek başına gösterirken, şekil 2-c bunları Gantt şeması olarak birlikte göstermektedir.



Şekil 2-a: DVFS Methodu Kullanılmadan RM-PCP MCRTsim Simülasyon Sonuçları



Şekil 2-b: Her görev için MCRTsim Zamanlama Simülasyonu



Şekil 2-c: MCRTsim Zamanlama Simülasyonu

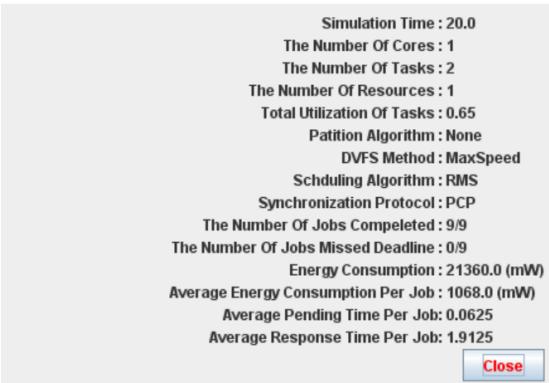
Şekiller kontrol edildiğinde,  $\tau_1$ 'in  $\tau_2$  görevini her geldiğinde önleyen sabit bir yüksek önceliğe sahip olduğu ve kritik bölüme girmesi gerektiğinde bloke olabileceği görülebilir.

Bu örnekte, herhangi bir güç azaltma tekniği kullanılmadan enerji tüketimi 21360 mW'dir. Buradaki işlemci, maksimum hız olan 1000 (%100) olan tek temel hızı üzerinde çalışmaktadır.

Ayrıca, işlemci Denklem 3'te  $\beta$  olarak gösterilen statik güç nedeniyle işlemci boştaiken bile enerji tüketir. İşlemciyi boştaiken kapatan DPM dahil edilirse, enerji tüketimi  $21230 - (7 \cdot 80) = 20800$  mW'a düşürülebilir.

Her zaman maksimum 1000 (%100) işlemci hızında çalışan MCRTsim'de sunulan MaxSpeed algoritması kullanıldığında tamamen aynı sonuçlar ve zamanlamalar elde edilecektir. Şekil 3-a, MaxSpeed kullanıldığında simülasyon sonuçlarını göstermektedir. Şekil 3-b ve 3-c, sırasıyla şekil 2-b ve 2-c ile tamamen aynıdır.

Simulation Results

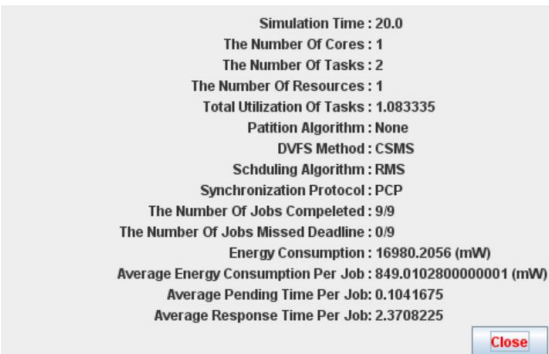


Şekil 3-a: MaxSpeed-RM-PCP MCRTSim Simülasyon Sonuçları

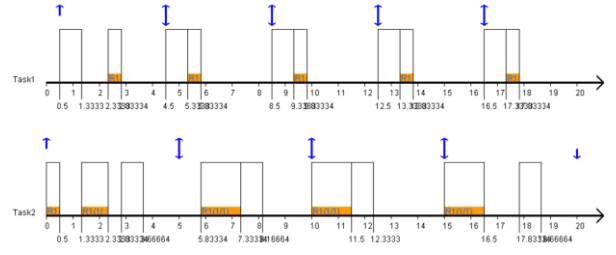
CSMS algoritması kullanılan bu konfigürasyonda kritik olmayan bölgelerdeki görevler 600 (%60) birim hızla çalıştırılırken, kritik bölümlerdeki görevler maksimum hız olan 1000 birim (%100) hızla çalıştırılmıştır. Kritik bölümlerde hız artırıldığı için enerji tüketimi de aynı oranda artmıştır ve 16980.2 (mW) olarak gözlemlenmiştir. Toplam çalışma süreside hızla ters orantılı olarak değişiklik göstermiştir.

Şekil 4-a, 4-b ve 4-c simülasyon sonuçlarını ve zamanlamaları göstermektedir. DPM de etkinleştirilirse, enerji tüketimi  $16980.2 - (4 \cdot 80) = 16660.2$  mW'a düşürülebilir.

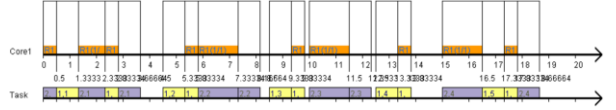
Simulation Results



Şekil 4-a: CSMS-RM-PCP MCRTSim Simülasyon Sonuçları



Şekil 4-b: Her görev için MCRTSim Zamanlama Simülasyonu



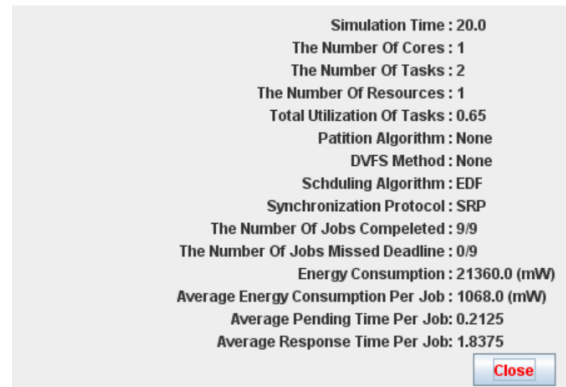
Şekil 4-c: MCRTSim Zamanlama Simülasyonu

### 3.2. DPM/DVFS metodunu EDF ve SRP ile Uygulama

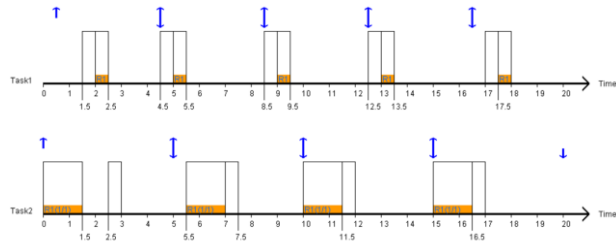
EDF'nin her işin son teslim tarihi ile ilgili dinamik bir önceliği vardır. Ayrıca, görev örneği (iş) geldiğinde SRP'de engelleme olmaktadır. Bu durum, şekil 5-b ve 5-c'de açıkça görülebilir.

EDF-SRP algoritması, DVFS desteklenmediğinde veya MaxSpeed tekniği kullanıldığında RM-PCP ile aynı enerjiyi tüketir.

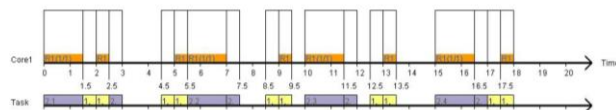
Simulation Results



Şekil 5-a: DVFS Methodu Kullanılmadan EDF-SRP MCRTSim Simülasyon Sonuçları



Şekil 5-b: Her görev için EDF-SRP MCRTSim Zamanlama Simülasyonu

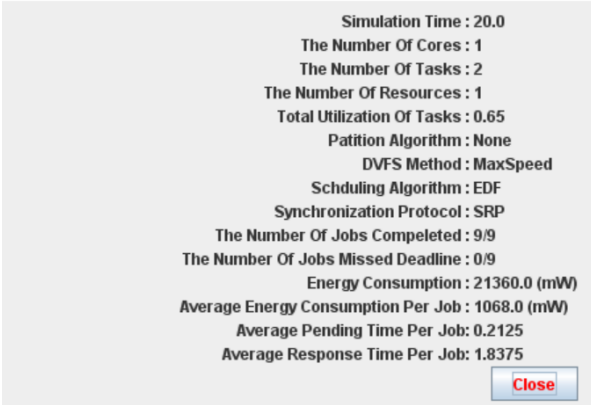


Şekil 5-c: EDF-SRP MCRTSim Zamanlama Simülasyonu

Oluşturulan konfigürasyon DVFS metodu kullanılmadan EDF zamanlayıcısı ve SRP kaynak erişim protokolü ile 1000 birim baz hızında çalıştırıldığında PCP için  $t=0$  anında önceliği düşük olan  $\tau_2$  görevi gelir, SRP'den dolayı  $t=0.5$  anında önceliği yüksek olan  $\tau_1$  görevi geldiği zaman işlem sırası  $\tau_1$  ile devam eder. DVFS metodunun kullanılmadığı bu durumda enerji tüketimi 21360 (mW) olarak gözlemlenmiştir. DPM etkinleştirilirse, enerji tüketimi de 20800 mW'a düşürülebilir.

MCRTsim üzerinde verilen konfigürasyon için zamanlama algoritması EDF, DVFS metodu olarak MaxSpeed, kaynak erişim protokolü SRP olmak üzere elde edilen sonuçlar ve grafik Şekil 6-a verilmiştir. Zamanlamalar şekil 5-b ve 5-c ile tamamen aynıdır.

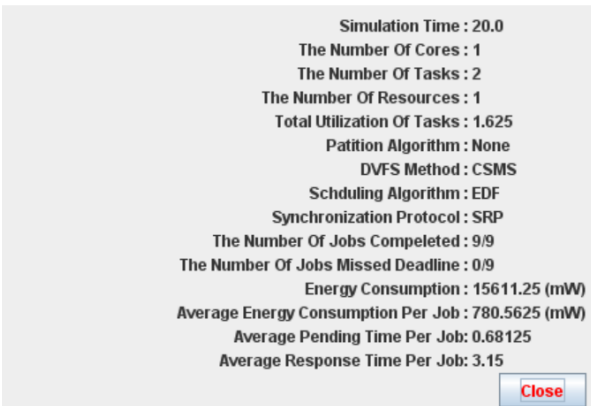
Simulation Results



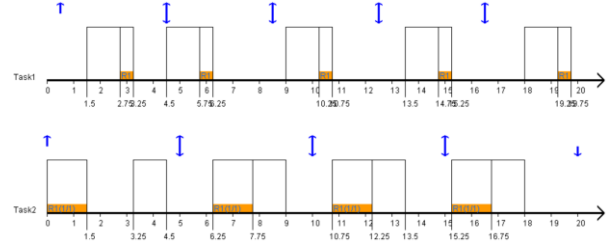
Şekil 6-a: MaxSpeed-EDF-SRP MCRTSim Simülasyon Sonuçları

MCRTsim üzerinde verilen konfigürasyon için zamanlama algoritması EDF, DVFS metodu olarak CSMS, kaynak erişim protokolü SRP olmak üzere elde edilen sonuçlar ve grafik Şekil 7-a, Şekil 7-b ve Şekil 7-c de verilmiştir.

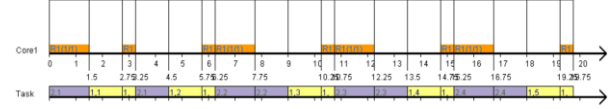
Simulation Results



Şekil 7-a: CSMS-EDF-PCP MCRTSim Simülasyon Sonuçları



Şekil 7-b: Her görev için EDF-SRP MCRTSim Zamanlama Simülasyonu



Şekil 7-c: EDF-SRP MCRTSim Zamanlama Simülasyonu

CSMS-EDF-SRP için enerji tüketimi 15611,25 mW'dir. DPM de kullanılırsa, enerji tüketimi  $15611,25 - (0,25 * 80) = 15591,25$  mW'a düşürülebilir. Burada kritik bölümler maksimum 1000 (%100) hızda, kritik olmayan bölümler ise 400 (%40) hızda yürütülür.

Simülasyon sonuçları ve zamanlama çizelgeleri karşılaştırıldığında kaynak erişim protokolünü değiştirince sonuçların değişmediği sadece görevlerin kritik bölümde çalışma sırasının değiştiği görülmektedir.

Aynı konfigürasyon üzerinde farklı DVFS Metotları çalıştırıldığında elde edilen sonuçlar Tablo 1'de gösterilmiştir.

Tablo1 de seçilen DVFS algoritmalarına göre elde edilen enerji tüketimi, çalışma süreleri ve çalıştırılan işlemci hızları gösterilmektedir. Hız artınca çalışma sürelerinin azaldığı fakat enerji tüketiminin arttığı görülmektedir. Karşılaştırılan algoritmalar göz önünde bulundurulduğunda CSMS algoritmasının zaman- hız tüketimi açısından daha optimal bir sonuç verdiği görülmüştür.

Verilen örnekte öncelik değişmediği için zamanlama algoritması olarak RM kullanıldığında da aynı sonuçlar elde edilmiştir.

**Tablo 1:** DVFS algoritmalarına ve verilen işlemci hızına göre elde edilen çalışma süreleri ve enerji tüketimi

DVFS Metodu	Zamanlama Algoritması	Çalışma Süresi		İşlemci Hızı		Enerji Tüketimi (mW)		Enerji Tüketimi Azaltılması (%)	
		Kritik Bölüm	Kritik Olmayan Bölüm	Kritik Bölüm	Kritik Olmayan Bölüm	DPM Olmadan	DPM	DPM Olmadan	DPM
Yok/MaxSpeed	RM	$\tau_1$ :0.50 $\tau_2$ :1.5	$\tau_1$ :0.50 $\tau_2$ :0.5	%100	%100	21360	20800	%0	%2.6
Yok/MaxSpeed	EDF	$\tau_1$ :0.5 $\tau_2$ :1.50	$\tau_1$ :0.5 $\tau_2$ :0.5	%100	%100	21360	20800	%0	%2.6
CSMS	RM	$\tau_1$ :0.5 $\tau_2$ :1.50	$\tau_1$ :0.833 $\tau_2$ :0.833	%100	%60	16980.2	16660.2	%20.5	%22
CSMS	EDF	$\tau_1$ :0.5 $\tau_2$ :1.50	$\tau_1$ :1.25 $\tau_2$ :1.25	%100	%40	15611.25	15591.25	%26.9	%27

#### 4. SONUÇ

Bu çalışmada tek işlemcili gerçek zamanlı sistemler üzerinde DVFS algoritmaları ele alınmış ve Java tabanlı MCRTsim uygulaması kullanılarak sonuçlar elde edilmiş ve sonuçlar karşılaştırılıp analiz edilmiştir.

Dinamik voltaj ve frekans ölçekleme (DVFS) tekniğinde amaç işlemcinin voltajını ve frekansını dinamik olarak ayarlayarak dinamik güç tüketimini azaltmayı amaçlamaktır. Elde edilen sonuçlar incelendiğinde de bu amacın gerçekleştirildiği ve aynı konfigürasyonun için enerji tüketiminin %27'ye kadar azaldığı gözlemlenmiştir.

Kaynak Erişim Protokolü olan PCP ve SRP arasında enerji tüketimi açısından bir farklılık görülmemiştir sadece kritik bölümlerde görevlerin çalışma sırası değişmiştir. Aynı şekilde zamanlama algoritmaları olan EDF ve RM arasında da enerji tüketimi açısından bir farklılık görülmemiştir.

#### KAYNAKLAR

Awadalla, M. & Elewi, A. (2016). Enhanced PSO approach for real time systems scheduling. *International Journal of Computer Theory and Engineering* 8(4), 285-289. <https://doi.org/10.7763/ijcte.2016.v8.1059>

Baker, T. P. (1991). Stack-based scheduling of realtime processes. *Real-Time Systems* 3(1), 67-99.

Buttazzo, G. C. (2011). *Hard real-time computing systems: predictable scheduling algorithms and applications*. Vol. 24. Springer Science & Business Media.

Cheng, A. M. K. & Ras, J. (2007). The implementation of the priority ceiling protocol in Ada-2005. *ACM SIGAda Ada Letters* 27(1), 24-39. <https://doi.org/10.1145/1274610.1274611>

Jejurikar, R., & Gupta, R. (2002). Energy aware edf scheduling with task synchronization for embedded real time systems. *Proc. of the Workshop on Compilers and Operating Systems for Low Power*.

Jejurikar, R. & Gupta, R. (2006). Energy-aware task scheduling with task synchronization for embedded real-time systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25(6), 1024-1037.

Laplante, P.A. & Ovaska, S.J. (2012). *Real-Time Systems Design and Analysis: Tools for the Practitioner*. 4th ed. Hoboken, NJ: JOHN WILEY & SONS.

Lipskoch, H., Albers, K., & Slomka, F. (2007). Fast Calculation of Permissible Slowdown Factors for Hard Real-Time Systems. *Proceedings of the 17th international conference on Integrated Circuit and System Design: power and timing modeling, optimization and simulation (PATMOS'07)*, Sweden, pp.495-504.

Liu, J. W. S., (2000). *Real-Time Systems*. Upper Saddle River, NJ: Prentice-Hall.

Liu, L. & Layland, J. W. (1973). Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *J. ACM*.

Palamut, S., Gönültaş, T., Elewi, A., & Avaroğlu, E. (2019). Task Scheduling Algorithms and Resource Access Protocols in Real Time Systems. In the *Proceedings of 2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. <https://doi.org/10.1109/IDAP.2019.8875974>

Pillai, P. & Shin, K.G. (2001). Real-time dynamic voltage scaling for low-power embedded operating systems. *SIGOPS Oper. Syst. Rev.* 35(5), 89-102. DOI:<https://doi.org/10.1145/502059.502044>

- Saad, E.M., Elewi, A., Shalan, M. & Awadalla, M. (2013). Energy and synchronization-aware mapping of real-time tasks on asymmetric multicore platforms. *International Journal of Computer Applications (IJCA)*, 75.11 (2013): 35-40. <https://doi.org/10.5120/13159-0932>
- Schmitz, M. T., Al-Hashimi, B. M., & Eles, P. (2005). System-level design techniques for energy-efficient embedded systems. *System-Level Design Techniques for Energy-Efficient Embedded Systems* (pp. 1-194). Springer US. <https://doi.org/10.1007/b106642>
- Sha, L. & Goodenough, J. (1990). Real-time scheduling theory and Ada. *IEEE Transactions on Computer*, Vol. 23, No. 4.
- Wu, J. & Huang, Y. (2017). MCRTsim: A simulation tool for multi-core real-time systems. In 2017 International Conference on Applied System Innovation (ICASI), 461-464.
- Yıldırım, M. K. , Süder, L. , Alkader, A. A. & Elewi, A. (2020). Çok İşlemcili Gerçek Zamanlı Sistemlerde Zamanlama Algoritmaları . *Bilgisayar Bilimleri ve Teknolojileri Dergisi* , 1 (2) , 42-50 . Retrieved from <https://dergipark.org.tr/en/pub/bibtet/issue/57253/788353>