SOUTH-WEST
UNIVERSITY
·NEOFIT RILSKY·
BLAGOEVGRAD, BULGARIA

VOLUME 2
2004

SCIENTIFIC
Research

ELECTRONIC
ISSUE

MULTITASK REGIME OF WORK FOR THE MICROPROCESSING SYSTEMS, AT THE AVAILABILY OF WIDE RANGE OF TIME CHANGING FOR THE MAINTANACE OF THE SEPARATE REQUESTS

Main assistant VLADIMIR KRUMOV GEBOV
Main assistant VELICHKA TEMELKOVA
South-west University of Bulgaria –2700 Blagoevgrad "H.Dimitar"2.
askon@ abv.bg , http://www.askon.hit.bg/

It is analyzed an existing condition at the development of configurations with one-chip-containing processor, that control processes in real time. There is a lack of information for the work of the systems in real time, at conditions of great dynamics and wide range of time changing for the maintenance of the requests, that occur at one and the some time. The organization that we offer has two main priorities: it leads to great incensement of the productivity of the processor in the concrete task and to freedom at the development and at the changing of the applied product.

I. Introduction

The interruptions in one microcomputer system come as result of defined happenings that occur asynchronous to each other. The main requirement to MSRT, (micro processing system for real time) are that output reaction of the system should be worked out in strictly defined time periods [1]. If we want to maintain the efficiently of the system at this conditions we should stick to this *multitasking* mode [2].
Multitasking in one-chip microprocessor system is pseudo-parallel or concurrent [2]. A real parallelism could exist only in multiprocessing systems. This demand the necessity of working out the following tasks:

- Switching of the tasks with minimal system costs
- Small size of the system program (kernel)
- Dispatching of the interruptions by
- Minimal interval when the interruptions are forbidden
- Quick reaction of the out interruptions
- System primitives for synchronization and interaction of the tasks
- Real time work
- Memory control

In one-processoring systems the processor is the only active recourse that is distributed by the kernel. The strategy for non-preemptive dispatching (strategy for *without pushing out*) is based on the principal for hysteresis where the changes are repressed. The recourses coast for maintenance of the switching define the main system recourses of the multitasking

According the hysterecis principal the strategy for planning *without pushing out* limit minimal number of the switching. Unnecessary system coast are avoided by synchronization of the next task.
It is found out that increasing of the tasks and high dispersion of the times for physical and logical processing could not apply this principal.

II. MAIN CONCEPTIONS AND CLASSOFICATION, CONNECTED TO THE TASKS

The task is an active object for decomposition of the system and is used as base logical issue for the distribution of the resources.
In each moment of the existing of the applied system each task may be: :

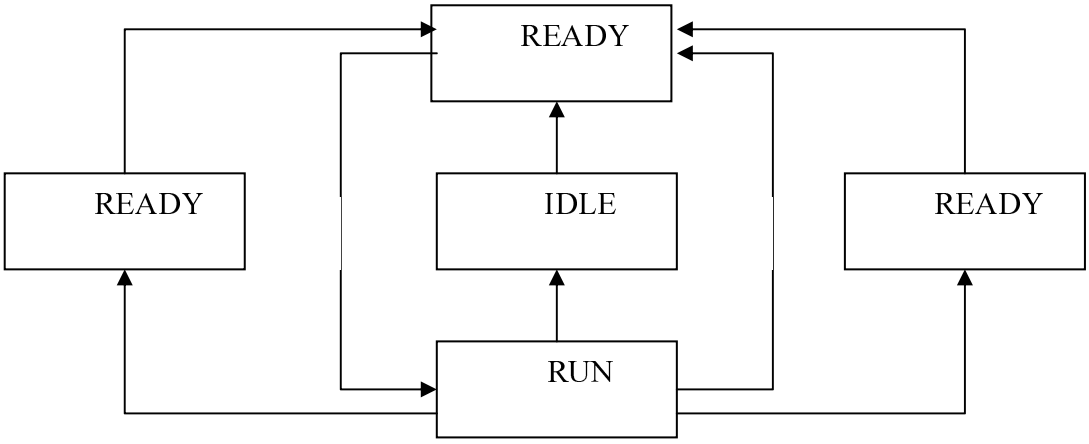

Fig. 1 Graphic of the condition of the task modulation under the control of kernel
1.Establishment 2.Re-establishment 3.Activation 4.Re-activation
5.Blocking 6.Unblocking 7.Holding 8. Admission

- Passive (IDLE) – only one resource, that is occupied by the task, is the area for the code of the program in its memory;

- Ready (READY) – The task has all necessary resource for its own the implement except of the processor;

- Active (RUN) The task is running at the moment;

- Blocked (BLOCK) – The task expects the beginning of a definite event or releasing of a resource.

Turning to one or another condition is controlled by the system organist- kernel.

The main problem at the solution of the problems, connected to the maintenance of the separate requests, is the frequency of their appearance and the times of their maintenance. At Fig.2 are classified the requests according to their frequencies and times of maintenance

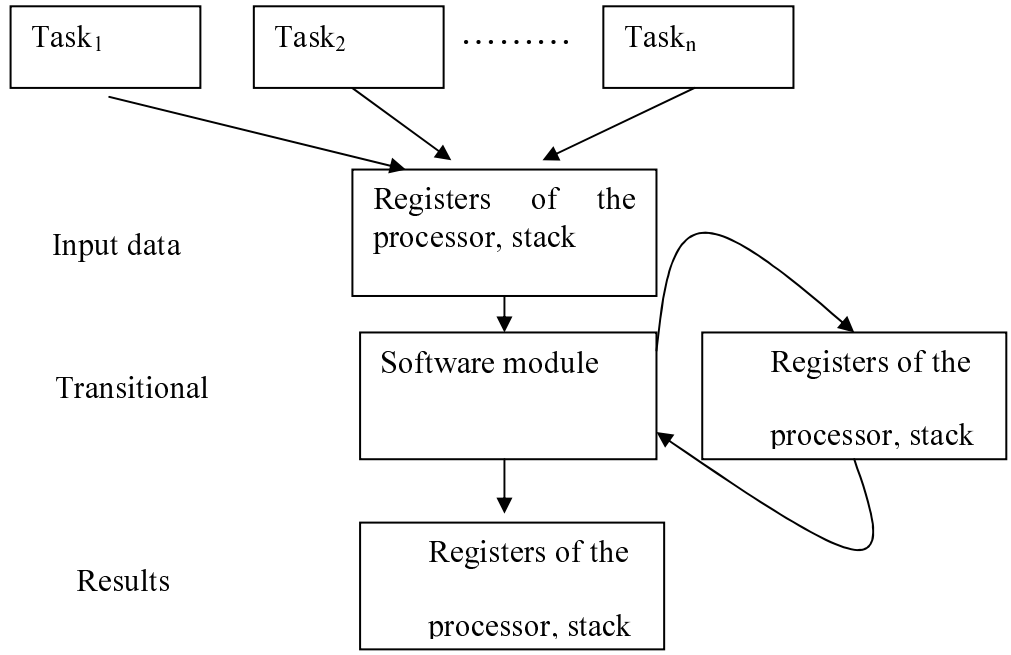

Fig.3. Data exchange between the queries for interruptions and module for arithmetical or logical proceeding.

Because of their different frequencies of appearance they are defined as it follows:
Controlled (CONTROL). Frequency of the appearance is controlled by kernel;

- Fixed frequency (FIX). The frequency is fixed and is activated by the relevant interruption;
- Variable frequency of appearance (VARIABLE). The variable frequency of appearance is defined with probability of appearance at one definite time period $p(\tau)$.

The maintenance of the requests is connected with two main times:

- Urgent time for maintenance (Exigent);
- Full time for maintenance of the requests (Execution).

According the hysterezis principal the strategy for planning *without pushing out* limit minimal number of the switching. Unnecessary system coast are avoided by synchronization of the next task. It is found out that increasing of the tasks and high dispersion of the times for physical and logical processing could not apply this principal.

III. DISCUSSION

Because of all these considerations in the most cases the authors recommend to use higher frequency interruption by equal time, and standard maintenance of the entry signals during time [2].

The analyze of this method shows that the fowling problems and discomforts are coast at interruption processing. .:

- The time for proceeding different tasks is equal distributed, and it doesn't depend on the type of the interruption. Quantization of the time in equal periods -t_k considers with the highest of the outside interruption, that causes time lost:

$$T_l = n * t_k - \sum_{i=1}^{n} t_{task}$$ -time for pause of the processor

(4.1-6)

$$T_p = \sum_{i=1}^{n} t_k$$ -Period of interruption at equal intervals, where $t_k > \max(t_{i\,task})$

- Increasing of the dispersion $\sigma(t_{i,task})$ causes incensement for time lost .
- The change of type and characters or adding the new interruption causes change of the time distribution t_k as well the period of interruption T_p.

All this requires necessity of development of the new method for the solution of this problem, which supposes solution of some other organization facilities for dependability of the system (Fig3):

- Organization of software module that work in parallel mode together with some other queries. The usage of the registers and stack of processor for data exchange output, transitional results and final;
- Processing of the buffer from RAM, registers of the processor, registers that control the interruptions are available in the same time from a few modules maintain interruptions;
- For demonstration of this method we could accept that two modules which maintain the interruption from the entries of processor IC2, IC3 change one buffer, named as BUFER.

```
LDAA   #I31F
STAA   TFLG1+REGBAS   clear the interrupt flag
LDX    #TMSK2+REGBAS
LDAA   0,X
PSHA                  memorizing in stack TMSK2
BCLR 0,X RTI          forbidden of the interruption RTI
LDX    #TMSK1+REGBAS
BCLR   0,X I21I|I31I  forbidden of the interruption IC1,IC2
BSET   BUFER IC1mask  change IC1mask from BUFER
```

Examples for maintain the interrupt from IC3.

Note: row 1,2 could not be replaced by with instruction for bit manipulation at data transfer by RS 485, or between massive is necessary to synchronize at fixed moment of the time.

IV. CONCLUSION

The organization of the interrupts by this method helps us to order and process the tasks that maintain the queries for interrupts. This method enables to improve the productivity of the processor, dependability of the work the algorithms as well as to decrease the time for projecting applying and at most the time that is consumed at the change of already existing system. This method enables to succeed the maximal produce with universal one-chip eight bit microprocessors.